122x
0

632

AD A074535

AD 074535

R and D TARADCOM
LABORATORY
TECHNICAL REPORT

NO. 12444

20020725104

PARAMETRIC ENGINEERING SYSTEM DEFINITION MODEL

VOLUME II

APPENDIX C (FORTRAN LISTINGS)

July 1979

Contract DAAK30-78-C-0059

by S. Spaulding, A. Weintraub,
F. Cioch, J. Lenz

Vector Research, Incorporated
P.O. Box 1506
Ann Arbor, Michigan 48106

AD 42907

U.S. ARMY TANK-AUTOMOTIVE
RESEARCH AND DEVELOPMENT COMMAND
Warren, Michigan 48090

Approved for public release; distribution unlimited.

APPENDIX C (FORTRAN LISTINGS)

TABLE OF CONTENTS

TABLE OF CONTENTS (concluded)

C.1: INTRODUCTION

This appendix contains the FORTRAN program listings for the COMPEND model. The listings are organized as follows:

- Section C-2 contains the main program, a listing of all labeled COMMON blocks (all other subprogram listings contain only the common block labels), and the BLOCKDATA subprogram.
- Section C-3 contains initialization and input processing routines.
- Section C-4 contains the main solution routines, including those used to estimate vehicle engineering and performance parameters.
- Section C-5 contains the output routines.
- Section C-6 contains utility routines.

C.2: MAIN PROGRAM, LABELED COMMONS AND BLOCK DATA

This section contains the following listings:

- The main program.
- A consolidated listing of all labeled COMMON blocks (subprogram listings in other sections of this report contain only the block labels for COMMON blocks used in that subprogram).
- The BLOCK DATA subprogram.

MAIN PROGRAM

```
ISN
      C          COMBAT VEHICLE DEFININITION MODEL
      C
      C       THIS PROGRAM PRODUCES A DESCRIPTION OF AN ARMORED COMBAT
      C    VEHICLE OS A SPECIFIED CLASS WHICH MEETS THE USER'S SPECIFICATIONS.
      C    AS PART OF HIS SPECIFICATIONS THE USER MAY INDICATE VARIOUS
      C    PERFORMANCE CRITERIA THAT THE VEHICLE MUST MEET.  HE MAY ALSO
      C    INDICATE VARIOUS ENGINEERING PARAMETERS DESCRIBING ITS COMPONENTS.
      C    FURTHERMORE HE MAY INDICATE WHICH SPECIFIC COMPONENTS ARE TO BE
      C    USED IN THE CONCEPT VEHICLE.  THE PROGRAM USES A DESCRIPTION OF
      C    WHAT COMPONENT TYPES AND HOW MANY OF EACH ARE NEEDED TO PRODUCE A
      C    COMBAT VEHICLE IN THE SPECIFIED CLASS.  IT ALSO READS IN A LIST OF
      C    ALTERNATIVE COMPONENTS OF EACH TYPE IT CAN USE IN ASSEMBLING THE
      C    CONCEPT VEHICLE.  IT THEN USES A GENERATE-AND-TEST PROCEDURE TO
      C    ASSEMBLE COMPONENTS WHICH WILL PRODUCE A FINISHED VEHICLE MEETING
      C    THE USER'S SPECIFICATIONS.
      C       FOR INPUT AND OUTPUT THE PROGRAM USES THE FOLLOWING DATA SETS:
      C
      C         DSRN       I/O          FILE DESCRIPTION
      C          1          I         DATA USED IN DESCRIBING FUNCTIONAL RELNS
      C          2          I         EXISTING VEHICLE FILE
      C          3          I         PROTOTYPE VEHICLE FILE
      C          4          I         ALTERNATIVE COMPONENT FILE
      C          5          I         USER SPECIFICATION FILE
      C          6          O         ERROR MESSAGES
      C          7          O         OUTPUT FILE
      C          8          O         ECHO OF USER SPECS & DATA HEADERS
      C
  1           LOGICAL ALTV
  2           DATA ALTV/.FALSE./
      C
      C    PERFORM INITIALIZATIONS
  3           CALL INIT
      C
      C    READ & STORE FUNCTIONAL RELATION DATA FILE
  4           CALL INFUNC
```

```
                                        MAIN PROGRAM
ISN
        C
        C  READ & STORE PROTOTYPE VEHICLE DEFINITION FILE
  5           CALL INPROT
        C
        C  READ & STORE ALTERNATIVE COMPONENT FILE
  6           CALL INALTC
        C
        C  READ & STORE USER SPECIFICATIONS IMAJING EXTERNAL FORM
  7           CALL INSPEC
        C
        C  EXTRACT ANY INFORMATION NEEDED FROM EXISTING VEHICLE FILE TO
        C  PRODUCE FINAL INTERNAL DESCRIPTION OF USER SPECIFTCATIONS
  8           CALL INVEHC
        C
        C  GENERATE "BEST" CONCEPT VEHICLE USING A BACKTRACK PROGRAMMING
        C  APPROACH
  9           CALL GENVEH
        C
        C  OUTPUT A DESCRIPTION OF THIS CONCEPT VEHICLE
 10           CALL OUTVEH
        C
 11           STOP
 12           END
```

COMMON BLOCKS

```
 COMMON /VEH/ VEHCLE(72,3),HULL(48,9),TURRET(42,9),MAINGN(44,9),
+     MACGUN(40,10),MISGUN(40,7),ADGUN(40,6),RANGER(28,7),
+     SENSOR(32,9),STABLE(30,6),GUNCTL(28,9),AMMO(30,30),
+     AMMOC(26,9),ENGINE(40,8),TRANSM(32,8),FINLDR(28,10),
+     ROADWH(36,10),SPRING(30,10),TRACK(34,8),SKIRT(28,8),
+     FUEL(28,10),FUELC(26,9),CREW(26,5),CARGO(26,5),ELECTR(26,6),
+     COMMO(28,9),FIREX(28,10),ENVIRC(26,9),DIAGNS(26,5),
+     SIGSUP(26,24),SMOKEG(26,10),EWSYS(26,10),RELN(3,50),
+     ISPECV,ICANDV,IALTV
 DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
+     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
+     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
+     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
+     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
+     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
+     IFLECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
+     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
+     IRELN(3,50),VFILE(1),IVFILE(1)
 EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
+     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
+     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
+     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
+     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
+     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
+     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
+     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
+     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
+     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
+     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 COMMON/VPROTO/ FLDNAM(1049),RECNAM(33),DFAULT(1049),
+     CMPCOD(31),ICMPPT(31),ICMPRK(31),IFILPT(33),IFLDPT(34),
+     IFLDTP(1049),IFLDVL(1049),LSTOUT,LSTREC(33),MAXNUM(31),
+     MINNUM(31),NALTCS(31),NFLDS(33),NRECS(33),OUTNAM(7,450),
```

COMMON BLOCKS

```
+     UNITS(1049),ICMATR,IEPARS,IPPARS,IOUTF(450)
 REAL*8 FLDNAM,RECNAM,DFAULT
 COMMON /JPARS/
+     JACCEL,JACCUR,JACOST,JADGUN,JAI,JAMMO,JAMMOC,JAMOTP,
+     JANGLE,JARMTP,JBETAF,JBETAR,JBI,JBORE,JBRAKE,JCALIB,
+     JCARGO,JCMPLX,JCOMMO,JCOOLR,JCOVER,JCREW,JCTLSY,JDAMP,
+     JDELD,JDIAGN,JDPRES,JDTRNG,JDWID,JDWNUP,JD1,JD2,JD3,
+     JECOST,JEFFIC,JEFPWR,JELEC,JELEVN,JEMOB,JENGIN,JENVR,
+     JEPROT,JERAMD,JEWSYS,JFCAP,JFGAL,JFINDR,JFIREX,JFRATE,
+     JFUEL,JFUELC,JFUELE,JFUELN,JGAMD,JGAMU,JGHPTN,JGPRES,
+     JGUIDE,JHMCVE,JHP,JHULL,JK1,JLOAD,JLOC,JMACHG,JMAG,
+     JMAING,JMANUF,JMATER,JMATUR,JMINWT,JMISL,JMODEL,JMUZLE,
+     JMXSPD,JNAT,JNCREW,JNFWDG,JNRDS,JNRDWH,JNRETN,JNRVSG,
+     JNUM,JOCOST,JOCIAM,JOHT,JPERF,JPERSN,JPHITM,JPHITS,
+     JPK,JPSLOP,JRANGA,JRANGE,JRCOST,JRELIB,JRNGSY,JROADW,
+     JRTIME,JSADJ,JSENSR,JSHPTN,JSIGSP,JSKHT,JSKIRT,JSKTHK,
+     JSLOPE,JSMCKE,JSPD1,JSPD2,JSPRNG,JSTBSY,JSTEMP,JSTEM2
 COMMON /JPARS1/
+     JSTIME,JSUSP,JT,JTBLEN,JTFIRS,JTFIR1,JTHFL,JTHFU,
+     JTONFT,JTRACK,JTRAD,JTRANS,JTRATE,JTRNRQ,JTTB,
+     JTTF,JTTS,JTTU,JTURET,JTYPE,JTYPEE,JVACOS,JVCMPX,JVHT,
+     JVLEN,JVMATR,JVMMBF,JVOCOS,JVOL,JVOPHR,JVVOL,JVWID,
+     JVWT,JWDEPN,JWDEPP,JWHTRV,JWT,JXH,JXTP,JX1,JX2,
+     JX3,JX4,JX5,JX6,JX7,JX8,JYEAR,JYGC,JYH,JY2,JY20,JY21,
+     JY22,JY23,JY3,JY4,JY5,JY6,JY7,JY8,JY9,JZH,JZ1,JZ10,
+     JZ11,JZ12,JZ13,JZ14,JZ15,JZ2,JZ3,JZ4,JZ5,JZ7,JZ8,JZ9,
+     JID,JNEXT,JRELOP,JSCORE,JVALUE,JKEY,JLT,JPENTR,
+     JSIGMA,JSIGM2,J30SPD
 COMMON /NPARS/
+     NARMC,NARMC2,NARMTP,NADIMS,NARMTH,NARRI,
+     NARLAB,NARRS,NASPEC,NCOMAT,NCSPCS,NCMPTP,
+     NCMPT1,NCMPS,NCMPZL,NDAYNT,NDETTP,NDISPR,NESPCS,NEVALC,
+     NFLDRC,NGEAR,NGRADE,
+     NWHTRV,NRDWHL,NORDT2,NORDTP,NOUT,NOUTWD,NPSPCS,
+     NRANGE,NRECTP,NRCTP1,NRELOP,NSHPTN,NSTABL,NSMTGT,NTERTP,
+     NTKLEN,NENGTP,NTYPES,NUTRNI,NUTRNS,NVEHWT,NXY
```

COMMON BLOCKS

```
 COMMON /OUTPUT/ XNAT(32),XMANUF(32),XMODEL(32),NUMBER(32),
+     WEIGHT(32),PCWT(32),VOLUME(32),PCVOL(32),DWNUP(32),
+     MATUR(32),ICMPLX(32),RELIAB(32),INDX(32),NPROD(32),
+     RDTIME(32),KOUT(32),FMCWGT,DF(8,31)
 LOGICAL*1 DF
 REAL*8 XNAT,XMANUF,XMODEL
 COMMON /AUX/ BL8,DASHES,BLS,IRELTP(15),ITYPES(4),MULT,NULL,
+     PLUS,RELNAM(15),SMINUS,TUNITS(3,50)
 REAL*8 BL8,DASHES
 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 COMMON /ARRAYS/ DETRNG(3,2,9),NATION(160),MANUF(160),
+     MODEL(160),PHITM(3,2,3,3),PHITS(3,2,3,3),
+     PK(3,2,3,3),PPENET(3,2,3,3),TRNRAD(4,3),SIGMA(2,3,3,16),
+     SIGMA2(2,3,3,16),ARRLAB(5,75),IARDEF(9,28),IARDF2(6,13)
 REAL*8 MANUF,MODEL,NATION
 DIMENSION ARRAY(1)
 EQUIVALENCE (ARRAY(1),DETRNG(1))
 COMMON /JPARS2/ JARSIZ,JARLST,JARNUM,JARRPT,
+     JDNAME,JFLDRC,JLEVPT,JNLEVL,JRECTP
 COMMON /SPECS/ ATRENG(200),ATRPER(34),VEHCAT,VEHCMP(31),
+     VEHENG(200),VFHPER(34),VEHWT,VEHEVL(5),NUMCMP(31),
+     TYPCMP(31),INDCMP(31),LSTCMP,TYPENG(200),RELENG(200),
+     MULENG(200),VL1ENG(200),ADDENG(200),VL2ENG(200),INDENG(200),
+     LSTENG,RELPER(34),MULPER(34),VL1PER(34),ADDPER(34),
+     VL2PER(34),LSTPER
 REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
 COMMON /SPECS2/ VEHS(10),ICMPS(31),ICOLS(10),IRECS(6,250),
+     JCOL,JDONE,JNXT,JSPEC,JTYP,JVEH,LSTVEH,LVEH,NIRECI,NVEHS,NVR
 REAL*8 VEHS
 COMMON /DATA/ ACCEL(4,5),CMPWTS(31),DENSTY(8),
+     MAXWHL,MINWHL,PHHULL,PHULLP(4,3,2,2,4),PKDAT(3,3,2,3),
+     PTURRP(4,3,2,2,4),RARMTH(4),RLTMAX,RLTMIN,
+     RSHPTN(4),RVEHWT(5),RTKLEN(4),RWHTRV(4),
+     SIGMVF(2,3,3,3),SIGMVT(2,3,3,2),
+     SLOPE(4,5),SPEED(4,2),VRIDE(4,5,4,3,2),WIDMAX,XMPG(4,5,4)
 COMMON /DATA2/ RDISPR(3)
```

BLOCK DATA

```
      BLOCK DATA
C
      COMMON /VEH/ VEHCLE(72,3),HULL(48,9),TURRET(42,9),MAINGN(44,9),
     +     MACGUN(40,10),MISGUN(40,7),ADGUN(40,6),RANGER(28,7),
     +     SENSOR(32,9),STABLE(30,6),GUNCTL(28,9),AMMO(30,30),
     +     AMMOC(26,9),ENGINE(40,8),TRANSM(32,8),FINLDR(28,10),
     +     ROADWH(36,10),SPRING(30,10),TRACK(34,8),SKIRT(28,8),
     +     FUEL(28,10),FUELC(26,9),CREW(26,5),CARGO(26,5),ELECTR(26,5),
     +     COMMO(28,9),FIREX(28,10),ENVIRC(26,9),DIAGNS(26,5),
     +     SIGSUP(26,24),SMOKEG(26,10),EWSYS(26,10),RELN(3,50),
     +     ISPECV,ICANDV,IALTV
      DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
     +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
     +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
     +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
     +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
     +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
     +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
     +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
     +     IRELN(3,50),VFILE(1),IVFILE(1)
      EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
     +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
     +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
     +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
     +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
     +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
     +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
     +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
     +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
     +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
     +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
     +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
     +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
      EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
      COMMON/VPROTO/ FLDNAM(1049),RECNAM(33),DFAULT(1049),
     +     CMPCOD(31),ICMPPT(31),ICMPRK(31),IFILPT(33),IFLDPT(34),
```

BLOCK DATA

```
+     IFLDTP(1049),IFLDVL(1049),LSTOUT,LSTREC(33),MAXNUM(31),
+     MINNUM(31),NALTCS(31),NFLDS(33),NRECS(33),OUTNAM(7,450),
+     UNITS(1049),ICMATR,IEPARS,IPPARS,IOUTF(450)
 REAL*8 FLDNAM,RECNAM,DFAULT
 COMMON /JPARS/
+     JACCEL,JACCUR,JACOST,JADGUN,JAI,JAMMO,JAMMOC,JAMOIP,
+     JANGLE,JARMTP,JBETAF,JBETAR,JBI,JBORE,JBRAKE,JCALIB,
+     JCARGO,JCMFLX,JCOMMC,JCOOLR,JCOVER,JCREW,JCTLSY,JDAMP,
+     JDELD,JDIAGN,JDPRES,JDTRNG,JDWID,JDWNUP,JD1,JD2,JD3,
+     JECOST,JEFFIC,JEFPWR,JELEC,JELEVN,JEMOB,JENGIN,JENVR,
+     JEPROT,JERAMD,JEWSYS,JFCAP,JFGAL,JFINDR,JFIREX,JFRATE,
+     JFUEL,JFUELC,JFUELE,JFUELN,JGAMD,JGAMU,JGHPTN,JGPRES,
+     JGUIDE,JHMCVE,JHP,JHULL,JK1,JLOAD,JLOC,JMACHG,JMAG,
+     JMAING,JMANUF,JMATER,JMATUR,JMINWT,JMISL,JMODEL,JMUZLE,
+     JMXSPD,JNAT,JNCREW,JNFWDG,JNRDS,JNRDWH,JNRETN,JNRVSG,
+     JNUM,JOCOST,JOCIAM,JOHT,JPERF,JPERSN,JPHITM,JPHITS,
+     JPK,JPSLOP,JRANGA,JRANGE,JRCOST,JRELIB,JRNGSY,JROADW,
+     JRTIME,JSACJ,JSENSR,JSHPTN,JSIGSP,JSKHT,JSKIRT,JSKTHK,
+     JSLOPE,JSMOKE,JSPD1,JSPD2,JSPRNG,JSTBSY,JSTEMP,JSTEM2
 COMMON /JPARS1/
+     JSTIME,JSUSP,JT,JTBLEN,JTFIRS,JTFIR1,JTHFL,JTHFU,
+     JTONFT,JTRACK,JTRAD,JTRANS,JTRATE,JTRNRQ,JTTB,
+     JTTF,JTTS,JTTU,JTURET,JTYPE,JTYPEE,JVACOS,JVCMPX,JVHT,
+     JVLEN,JVMATR,JVMMBF,JVOCOS,JVOL,JVOPHR,JVVOL,JVWID,
+     JVWT,JWDEPN,JWDEPP,JWHTRV,JWT,JXH,JXTP,JX1,JX2,
+     JX3,JX4,JX5,JX6,JX7,JX8,JYEAR,JYGC,JYH,JY2,JY20,JY21,
+     JY22,JY23,JY3,JY4,JY5,JY6,JY7,JY8,JY9,JZH,JZ1,JZ10,
+     JZ11,JZ12,JZ13,JZ14,JZ15,JZ2,JZ3,JZ4,JZ5,JZ7,JZ8,JZ9,
+     JID,JNEXT,JRELCP,JSCORE,JVALUE,JKEY,JLT,JPENTR,
+     JSIGMA,JSIGM2,J30SPD
 COMMON /NPARS/
+     NARMC,NARMC2,NARMTP,NADIMS,NARMTH,NARRI,
+     NARLAB,NARRS,NASPEC,NCOMAT,NCSPCS,NCMPTP,
+     NCMPT1,NCMPS,NCMPZL,NDAYNT,NDETTP,NDISPR,NESPCS,NEVALC,
+     NFLDPC,NGEAR,NGRADE,
+     NWHTRV,NRDWHL,NORDT2,NORDTP,NOUT,NOUTWD,NPSPCS,
```

BLOCK DATA

```
+     NRANGE,NRECTP,NRCTPL,NRELOP,NSHPTN,NSTABL,NSMTGT,NTERTP,
+     NTKLEN,NENGTP,NTYPES,NUTRNI,NUTRNS,NVEHWT,NXY
 COMMON /OUTPUT/ XNAT(32),XMANUF(32),XMODEL(32),NUMBER(32),
+     WEIGHT(32),PCWT(32),VOLUME(32),PCVOL(32),DWNUP(32),
+     MATUR(32),ICMPLX(32),RELIAB(32),INDX(32),NPROD(32),
+     RDTIME(32),KOUT(32),FMCWGT,DF(8,31)
 LOGICAL*1 DF
 REAL*8 XNAT,XMANUF,XMODEL
 COMMON /AUX/ BL8,CASHES,BLS,IRELTP(15),ITYPES(4),MULT,NULL,
+     PLUS,RELNAM(15),SMINUS,TUNITS(3,5))
 REAL*8 BL8,CASHES
 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 COMMON /ARRAYS/ DETRNG(3,2,9),NATION(160),MANUF(160),
+     MODEL(160),PHITM(3,2,3,3),PHITS(3,2,3,3),
+     PK(3,2,3,3),PPENET(3,2,3,3),TRNRAD(4,3),SIGMA(2,3,3,16),
+     SIGMA2(2,3,3,16),ARRLAB(5,75),IARDEF(9,28),IARDF2(6,13)
 REAL*8 MANUF,MODEL,NATION
 DIMENSION ARRAY(1)
 EQUIVALENCE (ARRAY(1),DETRNG(1))
 COMMON /JPARS2/ JARSIZ,JARLST,JARNUM,JARRPT,
+     JDNAME,JFLCRC,JLEVPT,JNLEVL,JRECTP
 COMMON /SPECS2/ VEHS(10),ICMPS(31),ICOLS(10),IRECS(6,250),
+     JCOL,JDONE,JNXT,JSPEC,JTYP,JVEH,LSTVEH,LVEH,NIRECI,NVEHS,NVR
 REAL*8 VEHS
 COMMON /DATA/ ACCEL(4,5),CMPWTS(31),DENSTY(8),
+     MAXWHL,MINWHL,PHHULL,PHULLP(4,3,2,2,4),PKDAT(3,3,2,3),
+     PTURRP(4,3,2,2,4),RARMTH(4),RLTMAX,RLTMIN,
+     RSHPTN(4),RVEHWT(5),RTKLEN(4),RWHTRV(4),
+     SIGMVF(2,3,3,3),SIGMVT(2,3,3,2),
+     SLOPE(4,5),SPEED(4,2),VRIDE(4,5,4,3,2),WIDMAX,XMPG(4,5,4)
 COMMON /DATA2/ RDISPR(3)
C
 DIMENSION KARDEF(9,14)
 EQUIVALENCE (KARDEF(1,1),IARDEF(1,14))
C
 DATA
```

```
     +     JACCEL/52/,JACCUR/26/,JACOST/4/,JADGUN/6/,JAI/25/,
     +     JAMMO/11/,JAMMOC/12/,JAMOTP/28/,JANGLE/27/,JARMTP/28/,
     +     JBETAF/33/,JBETAR/34/,JBI/25/,JBORE/29/,JBRAKE/56/,
     +     JCALIB/27/,JCARGO/23/,JCMPLX/9/,JCOMMO/25/,JCOOLR/27/,
     +     JCOVER/34/,JCREW/22/,JCTLSY/10/,JDAMP/27/,JDELD/38/,
     +     JDIAGN/28/,JDPRES/37/,JDTRNG/29/,JDWID/45/,JDWNUP/11/,
     +     JD1/29/,JD2/28/,JD3/27/,JECOST/22/,JEFFIC/26/,
     +     JEFPWR/18/,JELEC/24/,JELEVN/36/,JEMOB/19/,JENGIN/13/,
     +     JENVR/27/,JEPRCT/20/,JERAMD/21/,JEWSYS/31/,JFCAP/26/,
     +     JFGAL/26/,JFINDR/15/,JFIREX/26/,JFRATE/33/,JFUEL/20/,
     +     JFUELC/21/,JFUELE/29/,JFUELN/28/,JGAMD/37/,JGAMJ/36/,
     +     JGHPTN/41/,JGPRES/43/,JGUIDE/28/,JHMOVE/35/,JHP/26/,
     +     JHULL/1/,JK1/31/,JLOAD/30/,JLOC/17/,JMACHG/4/,
     +     JMAG/28/,JMAING/3/,JMANUF/14/,JMATER/26/,JMATUR/8/,
     +     JMINWT/38/,JMISL/5/,JMODEL/15/,JMUZLE/25/,JMXSPD/48/,
     +     JNAT/13/,JNCREW/25/,JNFWDG/27/,JNRDS/26/,JNRDWH/25/,
     +     JNRETN/26/,JNRVSG/28/,JNUM/12/,JOCOST/5/,JODIAM/42/,
     +     JOHT/44/,JPERF/26/,JPERSN/26/,JPHITM/39/,JPHITS/38/,
     +     JPK/40/,JPSLOP/57/,JRANGA/50/,JRANGE/51/,JRCOST/6/
      DATA
     +     JRELIB/10/,JRNGSY/7/,JROADW/16/,JRTIME/7/,JSADJ/27/,
     +     JSENSR/8/,JSHPTN/42/,JSIGSP/29/,JSKHT/26/,JSKIRT/19/,
     +     JSKTHK/27/,JSLOPE/53/,JSMOKE/30/,JSPD1/58/,JSPD2/59/,
     +     JSPRNG/28/,JSTBSY/9/,JSTEMP/32/,JSTEM2/33/,JSTIME/31/,
     +     JSUSP/17/,JT/28/,JTBLEN/26/,JTFIRS/32/,
     +     JTFIR1/31/,JTHFL/32/,JTHFU/31/,JTONFT/69/,JTRACK/18/,
     +     JTRAD/55/,JTRANS/14/,JTRATE/54/,JTRNRQ/30/,
     +     JTTB/39/,JTTF/36/,JTTS/37/,JTTU/38/,JTURET/2/,
     +     JTYPE/25/,JTYPEE/26/,JVACOS/65/,JVCMPX/62/,JVHT/35/,
     +     JVLEN/37/,JVMATR/61/,JVMMBF/63/,JVOCOS/66/,JVOL/3/,
     +     JVOPHR/64/,JVVOL/34/,JVWID/36/,JVWT/33/,JWDEPN/46/,
     +     JWDEPP/47/,JWHTRV/25/,JWT/2/,JXH/27/,
     +     JXTP/27/,JX1/39/,JX2/30/,JX3/29/,JX4/27/,JX5/35/,
     +     JX6/38/,JX7/29/,JX8/27/,JYEAR/16/,JYGC/68/,JYH/26/,
     +     JY2/39/,JY20/40/,JY21/41/,JY22/34/,JY23/35/,JY3/40/,
     +     JY4/41/,JY5/42/,JY6/34/,JY7/30/,JY8/31/,JY9/30/,
```

BLOCK DATA

```
+     JZH/25/,JZ1/32/,JZ10/36/,JZ11/37/,JZ12/34/,JZ13/33/,
+     JZ14/31/,JZ15/30/,JZ2/35/,JZ4/33/,JZ5/32/,JZ7/40/,
+     JZ8/43/,JZ9/44/,JZ3/45/
 DATA
+     JKEY/32/,JLT/67/,JPENTR/60/,JSIGMA/34/,JSIGM2/27/,
+     J30SPD/49/,
+     JID/24/,JNEXT/1/,JRELCP/2/,JVALUE/3/,JRECTP/1/,JFLDRC/2/,
+     JNLEVL/3/,JLEVPT/4/,JONAME/5/,JARSIZ/3/,JARLST/4/,JARRPT/5/,
+     JARNUM/6/,JSCORE/23/,JDONE/6/
 DATA
+     NARMC/16/,NARMC2/16/,NARMTP/8/,NADIMS/28/,NARMTH/4/,
+     NARLAB/75/,NARRS/13/,
+     NASPEC/2/,NCCMAT/25/,NCSPCS/31/,NCMPTP/31/,NCMPT1/32/,
+     NCMPS/160/,NCMPZL/12/,NDAYNT/2/,NDETTP/3/,NDISPR/3/,
+     NESPCS/200/,NEVALC/5/,NFLDRC/994/,NGEAR/4/,NGRADE/2/,
+     NIRECI/6/,NWHTRV/4/,NRDWHL/3/,NORDT2/2/,NORDTP/3/,
+     NOUT/450/,NOUTWD/7/,NPSPCS/34/,NRANGE/3/,NRECTP/33/,
+     NRCTP1/34/,NRELOP/15/,NSHPTN/4/,NSTABL/3/,NSMTGT/2/,
+     NTERTP/2/,NTKLEN/4/,NENGTP/4/,NTYPES/4/,NUTRNI/3/,
+     NUTRNS/50/,NVEHWT/5/,NXY/2/
 DATA RDISPR/1000.,2000.,3000./
 DATA NRECS/3,9,9,9,10, 7,6,7,9,6, 9,30,9,8,8, 10,10,10,8,8,
+     10,9,5,5,6, 9,10,9,5,24, 10,10,50/
 DATA NFLDS/72,48,42,44,40, 40,40,28,32,30, 28,30,26,40,32,
+     28,36,30,34,28, 28,26,26,26,26, 28,28,26,26,26, 26,26,3/
 DATA ISPECV/1/,ICANDV/2/,IALTV/3/
 DATA BLS/' '/,BL8/'        '/,MLLT/'*'/,NULL/-1/,PLUS/'+'/,SMINUS/'-'/,
+     DASHES/'--------'/
 DATA RELNAM/'LT','LE','EQ','NE','GE','GT','<','<=','=','¬=',
+     '>=','>',' ',' ',' '/
 DATA IPELTP/1,2,3,4,5,6,1,2,3,4,5,6,2,4,5/
 DATA ITYPES/'R','I','RA','C8'/
 DATA IARDEF/1,38,3,0,'RANG','E',' ',' ',' ',
+     1,38,2,0,'STAT',' VS ','MOVI','NG T','GT',
+     1,38,3,0,'CRDN','ANCE',' TYP','E',' ',
+     1,39,3,0,'RANG','E',' ',' ',' ',
```

BLOCK DATA

```
+       1,39,2,0,'STAT',' VS ','MOVI','NG T','GT',
+       1,39,3,0,'ORDN','ANCE',' TYP','E',' ',
+       1,40,3,0,'RANG','E',' ',' ',' ',
+       1,40,2,0,'ASPE','CT A','NGLE',' ',' ',
+       1,40,3,0,'ORDN','ANCE',' TYP','E',' ',
+       1,55,4,0,'GEAR',' ',' ',' ',' ',
+       1,60,3,0,'RANG','E',' ',' ',' ',
+       1,60,2,0,'ASPE','CT A','NGLE',' ',' ',
+       1,60,2,0,'ORDN','ANCE',' TYP','E',' '/
 DATA KARDEF/
+       4,34,2,0,'X OR',' Y S','IGMA',' ',' ',
+       4,34,3,0,'RANG','E',' ',' ',' ',
+       4,34,3,0,'ORDN','ANCE',' TYP','E',' ',
+       6,34,2,0,'X OR',' Y S','IGMA',' ',' ',
+       6,34,3,0,'RANG','E',' ',' ',' ',
+       6,34,3,0,'ORDN','ANCE',' TYP','E',' ',
+       8,27,2,0,'X OR',' Y S','IGMA',' ',' ',
+       8,27,3,0,'RANG','E',' ',' ',' ',
+       8,27,3,0,'ORDN','ANCE',' TYP','E',' ',
+       9,29,3,0,'DETE','CTIO','N CA','TEGO','RY',
+       9,29,2,0,'DAY ','VS N','IGHT',' ',' ',
+       11,27,2,0,'X OR',' Y S','IGMA',' ',' ',
+       11,27,3,0,'RANG','E',' ',' ',' ',
+       11,27,3,0,'ORDN','ANCE',' TYP','E',' '/
 DATA IARDF2/9,29,6,0,0,9,
+       0,13,2,0,0,160,
+       0,14,2,0,0,160,
+       0,15,2,0,0,160,
+       1,39,18,0,0,3,
+       1,38,18,0,0,3,
+       1,40,18,0,0,3,
+       1,60,18,0,0,3,
+       1,55,4,0,0,3,
+       4,34,18,0,0,8,
+       6,34,18,0,0,8,
+       8,27,18,0,0,8,
```

BLOCK DATA

```
+      11,27,18,0,0,8/
 DATA JNXT/1/,JTYP/2/,JSPEC/3/,JVEH/4/,JCOL/5/,
+      NVEHS/10/,NVR/250/
 DATA ICMPRK/3,5,7,8,9, 10,11,13,14,15, 20,16,17,18,19,
+      22,23,21,12,4,  6,1,2,24,25,  26,27,28,29,30,  31/
 DATA TUNITS/'KM','M',1000.,  'FT','IN',12.,  'M','FT',3.28,
+      'M','IN',39.37,  'IN','MM',25.4,  'RAD','DEG',57.3,
+      'MIN','SEC',60.,  'HR','MIN',60.,  'HR','SEC',3600.,
+      'TON','LB',2000.,  'TON','KG',907.,  'KG','LB',2.2,
+      'GAL','L',3.785,  'DEGC','DEGF',1.8,  'FT3','IN3',1728.,
+      'M/S','FT/S',3.28,  'MPH','KPH',1.61,  'M/S','MPH',2.24,
+      'MPH','FT/S',1.47/
 DATA FMCWGT/0.1/
 END
```

C.3: INITIALIZATION AND INPUT PROCESSING ROUTINES

This section contains listings of initialization and input processing routines. See section C-2 for listings of labeled COMMON blocks referred to in these listings.

```
ISN
 1         SUBROUTINE CSCORE
      C
      C    ROUTINE TO COMPUTE THE SCORE OF THE VARIOUS ALTERNATIVE COMPONENTS
      C AND TO ORDER THE COMPONENTS OF EACH TYPE IN DECREASING ORDER BY SCORE.
      C
 2         COMMON /VEH/
 3         DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +     IRELN(3,50),VFILE(1),IVFILE(1)
 4         EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5         EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6         COMMON /VPROTO/
 7         REAL*8 FLDNAM,RECNAM,DFAULT
 8         COMMON /JPARS/
 9         COMMON /JPARS1/
10         COMMON /NPARS/
11         COMMON /AUX/
```

```
                    INPUT ROUTINES -   CSCORE SUBROUTINE
ISN
12              REAL*8 BL8,DASHES
13              EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
14              COMMON /SPECS/
15              REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
       C
16              DIMENSION IRS(50),NEXT(50),SCORES(50)
       C
       C   ITERATE OVER COMPONENT TYPES
17              DO 50 ICMP=1,NCMPTP
       C
       C   GET ATTRIBUTES OF COMPONENT
18                 NF = NFLDS(ICMP+1)
19                 IOFF = IFILPT(ICMP+1)
       C
       C   ITERATE OVER COMPONENTS OF THAT TYPE
20                 IREC = ICMPPT(ICMP)
21                 LIST = NULL
22                 J = 0
23                 CALL NULIFY(NEXT,50)
24                 CALL ZERO(SCORES,50)
25     10          IF (IREC .EQ. NULL) GO TO 30
26                 IOFFR = IOFF + NF*(IREC-1) - 1
27                 SCORE = SUMPRD(VFILE(IOFFR+JEFPWR),VEHEVL,5)
28                 VFILE(IOFFR+JSCORE) = SCORE
       C
       C   INSERT NEW COMPONENT IN LIST IN DECREASING ORDER
29                 J = J + 1
30                 IRS(J) = IREC
31                 SCORES(J) = SCORE
32                 IF (J .NE. 1) GO TO 14
33     12             NEXT(J) = LIST
34                    LIST = J
35                    GO TO 20
36     14          IF (SCORE .GT. SCORES(LIST)) GO TO 12
37                 K = LIST
38     16          KP = K
```

```
ISN
39              K = NEXT(K)
40              IF (K .NE. NULL .AND. SCORE .LE. SCORES(K)) GO TO 16
41              NEXT(J) = K
42              NEXT(KP) = J
43    20        IREC = IVFILE(IOFFR+JNEXT)
44              GO TO 10
      C
      C  STRING TOGETHER COMPONENTS IN DECREASING ORDER BY THEIR SCORES
45    30        K = LIST
46              ICMPPT(ICMP) = LIST
47    32        IF (K .EQ. NULL) GO TO 50
48              IOFFR = IOFF + NF*(K - 1) - 1
49              IVFILE(IOFFR+JSCORE) = NEXT(K)
50              K = NEXT(K)
51              GO TO 32
      C
52    50        CONTINUE
53            RETURN
54            END
```

```
                INPUT ROUTINES -   INALTC SUBROUTINE
ISN
 1          SUBROUTINE INALTC
       C
       C    ROUTINE FOR READING AND STORING ALTERNATIVE COMPONENTS.
       C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +      IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6          COMMON /VPROTO/
 7          REAL*8 FLDNAM,RECNAM,DFAULT
 8          COMMON /JPARS/
 9          COMMON /JPARS1/
10          COMMON /NPARS/
11          COMMON /AUX/
12          REAL*8 BL8,DASHES
```

INPUT ROUTINES -   INALTC SUBROUTINE

```
ISN
13             EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
14             COMMON /ARRAYS/
15             REAL*8 MANUF,MODEL,NATION
16             DIMENSION ARRAY(1)
17             EQUIVALENCE (ARRAY(1),DETRNG(1))
18             CCMMON /JPARS2/
19             COMMON /SPECS/
20             REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
         C
21             REAL*8 V,CMP,CMP2,ATTR
22             DIMENSION INT(12),IRECS2(12),VALS(12),IVALS(12),CVALS(2,12)
         C
         C  READ VEHICLE CATEGCRY RECORD
23             READ (4,100) V
24       100   FORMAT (A8)
25             IF (V .NE. VEHCAT) CALL ERR(4,'INALTC',V,VEHCAT,0)
26             CMP2 = BL8
         C
         C  READ COMPONENT HEADER
27       2     READ (4,105,END=90) CMP,(INT(I),I=1,NCMPZL)
28       105   FORMAT (A8,2X,12(8X,I2))
         C  CHECK WHETHER HAVE ENCOUNTERED A NEW TYPE OF COMPONENT
29             IF (CMP .EQ. CMP2) GO TO 5
30             N = 0
31             CMP2 = CMP
         C
         C  DETERMINE HOW MANY NON-ZERO ALTERNATIVES THERE ARE
32       5     DO 10 I=1,NCMPZL
33                IF (INT(I) .EQ. 0) GO TO 15
34                IF (INT(I) .NE. N+I) CALL ERR(5,'INALTC',CMP,N+I, INT(I))
35       10       CONTINUE
36             N = N + NCMPZL
37             NA = NCMPZL
38             GO TO 20
         C
39       15    I = I - 1
```

```
                  INPUT ROUTINES -   INALTC SUBROUTINE
ISN
40            N = N + I
41            NA = I
      C
      C  IDENTIFY RECORD AND COMPONENT TYPES; STORE NUMBER OF ALTERNATIVES
      C  OF THIS TYPE COMPONENT
42    20      DO 25 IRECTP=1,NRECTP
43               IF (CMP .EQ. RECNAM(IRECTP)) GO TO 27
44    25         CONTINUE
45            CALL ERR(9,'INALTC',CMP,0,0)
46    27      ICMP = IRECTP - 1
47            IF (ICMP .GE. 1) NALTCS(ICMP) = N
      C
      C  DETERMINE OFFSET IN FILE IN WHICH  RECORDS STORED
48            IOFF = IFILPT(IRECTP)
      C  DETERMINE NUMBER OF FIELDS PER RECORD IN THIS FILE
49            NF = NFLDS(IRECTP)
      C  GET NA NEW COMPCNENT RECORDS AND QUEUE THEM ONTO THE CURRENT
      C  LIST
50            DO 30 IREC=1,NA
51               IR = NEWREC(IRECTP)
52               CALL STORE(INT(IREC),VFILE(IOFF),NF,JID,IR)
53               IRECS2(IREC) = IR
54               CALL QUEUE(IR,ICMPPT(IRECTP-1),IRECTP)
55    30         CONTINUE
      C
      C  READ COMPCNENT ATTRIBUTES
56    40      READ (4,110,END=90) ATTR
57    110     FORMAT (2X,A8)
      C  IS THIS THE END OF THIS SET OF ATTRIBUTES ?
58            IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 2
      C  FIND FIELD INDEX AND RECORD TYPE
59            IFLD = IATTR(ATTR,CMP,IRECTP)
      C  IF ATTRIBUTE NAME IS UNRECOGNIZED, HAVE AN ERROR
60            IF (IFLD .NE. NULL) GO TO 45
61               CALL ERR(6,'INALTC',CMP,ATTR,0)
62               GO TO 40
```

INPUT ROUTINES -   INALTC SUBROUTINE

```
ISN
      C
      C  DETERMINE TYPE CF ATTRIBUTE
 63   45    KFLD = IFLDPT(IRECTP) + IFLD
 64         ITYP = IFLDTP(KFLD)
      C  BACKSPACE RECORD AND READ VALUES USING FORMAT APPROPRIATE TO TYPE
 65         BACKSPACE 4
 66         DO 50 I=1,NTYPES
 67            IF (ITYP .EQ. ITYPES(I)) GO TO 55
 68   50       CONTINUE
 69         CALL ERR(10,'INALTC',CMP,ATTR,ITYP)
 70         GO TO 2
      C
      C  BRANCH APPROPRIATE TO TYPE
 71   55    GO TO (60,65,70,80),I
      C
      C  READ & STORE REAL VALUES
 72   60    READ (4,115) (VALS(J),J=1,NA)
 73   115   FORMAT (12X,12(F8.0,2X))
 74         BACKSPACE 4
 75         READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
 76         DO 61 J = 1,NA
 77            IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS)
           +   VALS(J) = FNULL
 78   61    CONTINUE
 79         DO 62 J=1,NA
 80            CALL STORE(VALS(J),VFILE(IOFF),NF,IFLD,IRECS2(J))
 81   62       CONTINUE
 82         GO TO 40
      C
      C  READ AND STORE INTEGER VALUES
 83   65    READ (4,120) (IVALS(J),J=1,NA)
 84   120   FORMAT (12X,12(I8,2X))
 85         BACKSPACE 4
 86         READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
 87         DO 67 J = 1,NA
 88            IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS)
 ISN
               +   IVALS(J) = NULL
  89    67       CONTINUE
  90             DO 68 J=1,NA
  91                CALL STORE(IVALS(J),VFILE(IOFF),NF,IFLD,IRECS2(J))
  92    68          CONTINUE
  93             GO TO 40
        C
        C  READ & STORE AN ARRAY OF REAL VALUES
        C
        C  DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
  94    70       DO 72 IARR=1,NARRS
  95                IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND. IARDF2(JFLDRC,IARR)
               +      .EQ. IFLD) GO TO 73
  96    72          CONTINUE
  97             CALL ERR(16,'INALTC',IRECTP,IFLD,ITYP)
        C
        C  NUMBER OF ELEMENTS IN ARRAY
  98    73       NV = IARDF2(JARSIZ,IARR)
        C  OFFSET IN FILE CONTAINING ARRAYS
  99             IOFFAR = IARDF2(JARRPT,IARR)
        C  GET NA VACANT ARRAYS OF TYPE IARR & STORE POINTERS TO THEM FROM THE
        C  CURRENT RECORD
 100             DO 75 J=1,NA
 101                K = IARDF2(JARLST,IARR) + 1
 102                IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INALTC',
               +      ATTR,IARDF2(JARNUM,IARR),IARR)
 103                IARDF2(JARLST,IARR) = K
 104                CALL STORE(K,VFILE(IOFF),NF, IFLD, IRECS2(J))
 105                IRECS2(J) = K
 106    75          CONTINUE
        C  READ NV VALUES FOR EACH OF THE NA ARRAYS AND STORE THEM
 107             DO 77 I=1,NV
 108                READ (4,115) (VALS(J),J=1,NA)
 109                BACKSPACE 4
 110                READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
 111                DO 76 J = 1,NA
```

```
                    INPUT ROUTINES -   INALTC SUBROUTINE
 ISN
112                  IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS)
            +        VALS(J) = FNULL
113    76          CONTINUE
114                DO 77 J=1,NA
115                  CALL STORE(VALS(J),ARRAY(IOFFAR),NV,I,IRECS2(J))
116    77            CONTINUE
117              GO TO 40
       C
       C  READ AND STORE CHARACTER INFORMATION (<= 8 CHARS)
118    80        READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
119    125       FORMAT (12X,12(2A4,2X))
       C
       C  DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
120              DO 82 IARR=1,NARRS
121                IF (IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 83
122    82          CONTINUE
123              CALL ERR(16,'INALTC',IRECTP,IFLD,ITYP)
       C
       C  OFFSET IN FILE CONTAINING CHARACTER STRINGS
124    83        IOFFCH = IARDF2(JARRPT,IARR)
       C  GET NA VACANT CHARACTER STRING RECORDS AND STORE POINTERS TO
       C  THEM FROM THE CURRENT RECORD.  ALSO STORE CHARACTER STRINGS IN
       C  THE CHARACTER STRING RECORDS.
125              DO 85 J=1,NA
126                K = IARDF2(JARLST,IARR) + 1
127                IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INALTC',
            +        ATTR,IARDF2(JARNUM,IARR),IARR)
128                IARDF2(JARLST,IARR) = K
129                CALL STORE(K,VFILE(IOFF),NF,IFLD,IRECS2(J))
130                CALL STORE(CVALS(1,J),ARRAY(IOFFCH),2,1,K)
131                CALL STORE(CVALS(2,J),ARRAY(IOFFCH),2,2,K)
132    85          CONTINUE
133              GO TO 40
       C
134    90        RETURN
135              END
```

```
                  INPUT ROUTINES -   INATTR SUBROUTINE
ISN
 1          SUBROUTINE INATTR(ATTR,IREC,ISPEC,IRECTP)
      C
      C        ROUTINE TO READ THE VALUE(S) OF AN ATTRIBUTE OF INTEREST
      C     ASSOCIATED WITH A GIVEN VEHICLE IN THE EXISTING VEHICLE FILE.
      C     THE ATTRIBUTE'S VALUE MAY BE A REAL, INTEGER, OR CHARACTER
      C     SCALAR (<= 8 CHARS) OR AN ARRAY OF REAL NUMBERS.
      C        THIS ROUTINE PROCESSES ATTRIBUTES WHICH WILL BE USED AS
      C     CONSTRAINTS AND STORED IN A RELN RECORD.  SUBROUTINE INATR2
      C     PROCESSES THOSE THAT ARE STORED AS CONSTANTS.
      C
      C     INPUT PARAMETERS:
      C        ATTR     ATTRIBUTE OF INTEREST
      C        IREC     RECORD OF IRECS CONTAINING INFORMATION ABOUT VEHICLE
      C                  OF INTEREST
      C        ISPEC    RECORD DESCRIBING USER SPECIFICATION
      C        IRECTP   TYPE OF RECORD THE ATTRIBUTE IS A FIELD OF
      C
 2          CCMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
           +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +     (FINLDR(1),IFINCR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
```

INPUT ROUTINES -   INATTR SUBROUTINE

```
 ISN
              +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
              +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
              +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
              +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
              +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
   5            EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
   6            COMMON /VPROTO/
   7            REAL*8 FLDNAM,RECNAM,DFAULT
   8            COMMON /JPARS/
   9            COMMON /JPARS1/
  10            COMMON /NPARS/
  11            COMMON /AUX/
  12            REAL*8 BL8,DASHES
  13            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
  14            COMMON /ARRAYS/
  15            REAL*8 MANUF,MODEL,NATION
  16            DIMENSION ARRAY(1)
  17            EQUIVALENCE (ARRAY(1),DETRNG(1))
  18            COMMON /JPARS2/
  19            COMMON /SPECS/
  20            REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
  21            COMMON /SPECS2/
  22            REAL*8 VEHS
          C
  23            REAL*8 ATTR
  24            DIMENSION CVALS(2,12),IVALS(12),VALS(12)
          C
          C  MUST BACKSPACE TO PCSITION RECORD POINTER APPROPRIATE FOR READING
  25            BACKSPACE 2
          C
          C  GET TYPE OF ATTRIBUTE
  26            IFLD = IATTR2(ATTR,IRECTP)
  27            KFLD = IFLDPT(IRECTP) + IFLD
  28            ITYP = IFLDTP(KFLD)
          C
          C  MARK ATTRIBUTE AS PROCESSED
```

INPUT ROUTINES -   INATTR SUBROUTINE

```
ISN
29              IRECS(JDCNE,IREC) = 1
       C
       C   GET COMPONENT IC IF THIS IS ENGINEERING PARAMETER CONSTRAINT
30              ID = 0
31              IF (IRECTP .GT. 1) ID = INDENG(ISPEC)
       C
       C   TRANSLATE RELOP TO A CANONICAL VALUE FROM 1 TO 6
32              DO 5 J=1,NRELOP
33                 IF (IRECTP .EQ. 1) REL = RELPER(ISPEC)
34                 IF (IRECTP .GT. 1) REL = RELENG(ISPEC)
35                 IF (REL .EQ. RELNAM(J)) GO TO 8
36     5           CONTINUE
37              IF (IRECTP .EQ. 1) CALL ERR(26,'INATTR','VEH',ATRPER(ISPEC),
              +      RELPER(ISPEC))
38              IF (IRECTP .GT. 1) CALL EPR(26,'INATTR',TYPENG(ISPEC),
              +      ATRENG(ISPEC),RELENG(ISPEC))
       C
39     8        KREL = IRELTF(J)
       C
       C   GET THE COLUMN OF INTEREST IN THE EXISTING VEHICLE FILE
40              IV = IRECS(JVEH,IREC)
41              ICOL = ICOLS(IV)
       C
       C   BRANCH ACCORDING TC CATATYPE
42              DO 10 I=1,NTYPES
43                 IF (ITYP .EQ. ITYPES(I)) GO TO 15
44     10          CONTINUE
45              CALL ERR(10,'INATTR',RECNAM(IRECTP),ATTR,ITYP)
46              RETURN
       C
47     15       GO TO (20,40,60,80),I
       C
       C   READ & STORE A REAL VALUE
48     20       READ (2,100) (VALS(I),I=1,ICOL)
49     100      FORMAT (12X,12(F8.0,2X))
50              VALUE = VALS(ICOL)
```

```
ISN
      C  TRANSFORM VALUE IF CALLED FOR BY USER SPECIFICATION
51           IF (IRECTP .GT. 1) GO TO 25
52               VAL = TRANSF(VALUE,MULPER(ISPEC),VL1PER(ISPEC),
            +      ADDPER(ISPEC),VL2PER(ISPEC))
53               GO TO 30
54     25    VAL = TRANSF(VALUE,MULENG(ISPEC),VL1ENG(ISPEC),
            +      ADDENG(ISPEC),VL2ENG(ISPEC))
      C  STORE VALUE AS A CCNSTRAINT OF THE RECORD
55     30    IR = NEWREL(IRECTP,IFLD,KREL,VAL,ID)
56           RETURN
      C
      C  READ & STORE AN INTEGER VALUE
57     40    READ (2,105) (IVALS(I),I=1,ICOL)
58     105   FORMAT (12X,12(I8,2X))
59           VALUF = IVALS(ICCL)
      C  TRANSFORM VALUF IF CALLED FOR BY USER SPECIFICATION
60           IF (IRECTP .GT. 1) GO TO 45
61               IVALUE = TRANSF(VALUE,MULPER(ISPEC),VL1PER(ISPEC),
            +      ADDPER(ISPEC),VL2PER(ISPEC))
62               GO TO 50
63     45    IVALUE = TRANSF(VALUE,MULENG(ISPEC),VL1ENG(ISPEC),
            +      ADDENG(ISPEC),VL2ENG(ISPEC))
64     50    IR = NEWREL(IRECTP,IFLD,KREL,IVALUE,ID)
65           RETURN
      C
      C  READ & STORE AN ARRAY OF REAL VALUES
      C
      C  DETERMINE WHICH ARRAY FILE THE INFORMATION WILL BE STORED IN
66     60    DO 62 IARR=1,NARRS
67               IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND.
            +      IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 63
68     62        CONTINUE
69           CALL ERR(16,'INATTR',IRECTP,IFLD,ITYP)
      C
      C  NUMBER OF ELEMENTS IN ARRAY
70     63    NV = IARDF2(JARSIZ,IARR)
```

```
                    INPUT ROUTINES -    INATTR SUBROUTINE
I SN
      C  OFFSET IN FILE CONTAINING ARRAYS
 71          IOFFAR = IARDF2(JARRPT,IARR)
      C  GET A VACANT ARRAY OF TYPE IARR
 72          K = IARDF2(JARLST,IARR) + 1
 73          IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATTR',ATTR,
     +           IARDF2(JARNUM,IARR),IARR)
 74          IARDF2(JARLST,IARR) = K
      C  GET A RELATION RECORD TO STORE THE CONSTRAINTS IN THIS FIELD IN
 75          IR = NEWREL(IRECTP,IFLD,KREL,K,ID)
      C  FOR EACH VALUE OF ARRAY SPECIFIED IN EXISTING VEHICLE FILE,
      C  READ, TRANSFORM, & STORE IT.
 76          DO 75 I=1,NV
 77             READ (2,100) (VALS(J),J=1,ICOL)
 78             VALUE = VALS(ICOL)
      C  TRANSFORM VALUE IF CALLED FOR BY USER SPECIFICATION
 79             IF (IRECTF .GT. 1) GO TO 65
 80                VAL = TRANSF(VALUE,MULPER(ISPEC),VL1PER(ISPEC),
     +                   ADDPER(ISPEC),VL2PER(ISPEC))
 81                GO TO 70
 82   65        VAL = TRANSF(VALUE,MULENG(ISPEC),VL1ENG(ISPEC),
     +            ADDENG(ISPEC),VL2ENG(ISPEC))
 83   70        CALL STORE(VAL,ARRAY(IOFFAR),NV,I,K)
 84   75        CONTINUE
 85          RETURN
      C
      C  READ & STORE A CHARACTER VALUE (<= 8 CHARS)
 86   80     READ (2,110) (CVALS(1,J),CVALS(2,J),J=1,ICOL)
 87   110    FORMAT (12X,12(2A4,2X))
      C  DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
 88          DO 82 IARR=1,NARRS
 89             IF (IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 83
 90   82        CONTINUE
 91          CALL ERR(16,'INATTR',IRECTP,IFLD,ITYP)
      C
      C  OFFSET IN FILE CONTAINING CHARACTER STRINGS
 92   83     IOFFCH = IARDF2(JARRPT,IARR)
```

```
                    INPUT ROUTINES -    INATTR SUBROUTINE
ISN
      C   GET A VACANT CHARACTER STRING RECORD
 93          K = IARDF2(JARLST,IARR) + 1
 94          IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATTR',ATTR,
            +      IARDF2(JARNUM,IARR),IARR)
 95          IARDF2(JARLST,IARR) = K
      C   CHECK THAT RELATION IS EQUALITY OR INEQUALITY (OTHER RELNS NOT
      C   APPROPRIATE FOR CHARACTER DATA).
 96          IF (KREL .NE. 3 .AND. KREL .NE. 4) CALL ERR(27,'INATTR',
            +      KREL,ATTR,IRECTP)
      C   GET A RELATION RECORD TO STORE THE CONSTRAINT ON THIS FIELD IN
 97          IR = NEWREL(IRECTP,IFLD,KREL,K,ID)
      C   STORE CHARACTER DATA IN CHARACTER RECORD
 98          CALL STORE(CVALS(1,ICOL),ARRAY(IOFFCH),2,1,K)
 99          CALL STORE(CVALS(2,ICOL),ARRAY(IOFFCH),2,2,K)
100          RETURN
101          END
```

INPUT ROUTINES -   INATR2 SUBROUTINE

```
ISN
 1          SUBROUTINE INATR2(ATTR,IREC,ISPEC,IRECTP)
      C
      C        ROUTINE TO READ THE VALUE(S) OF AN ATTRIBUTE OF INTEREST
      C     ASSOCIATED WITH A GIVEN VEHICLE IN THE EXISTING VEHICLE FILE.
      C     THE ATTRIBUTE'S VALUE MAY BE A REAL, INTEGER, OR CHARACTER
      C     SCALAR (<= 8 CHARS) OR AN ARRAY OF REAL NUMBERS.
      C        THIS ROUTINE PROCESSES ATTRIBUTES WHICH WILL BE
      C     STORED AS CONSTANTS IN A COMPONENT RECORD.  SUBROUTINE INATTR
      C     PROCESSES THOSE THAT ARE STORED AS CONSTRAINTS.
      C
      C     INPUT PARAMETERS:
      C        ATTR     ATTRIBUTE OF INTEREST
      C        IREC     RECORD OF IRECS CONTAINING INFORMATION ABOUT VEHICLE
      C                  OF INTEREST
      C        ISPEC    RECORD DESCRIBING USER SPECIFICATION
      C        IRECTP   TYPE OF RECORD THE ATTRIBUTE IS A FIELD OF
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
           +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
```

INPUT ROUTINES - INATR2 SUBROUTINE

```
ISN
          +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +     IENVIR(1)),(CIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
  5             EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
  6             CCMMON /VPROTO/
  7             REAL*8 FLDNAM,RECNAM,DFAULT
  8             COMMCN /JPARS/
  9             CCMMON /JPARS1/
 10             COMMON /NPARS/
 11             CCMMON /AUX/
 12             REAL*8 BL8,CASHES
 13             EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 14             CCMMON /ARRAYS/
 15             REAL*8 MANUF,MODEL,NATICN
 16             DIMENSION ARRAY(1)
 17             EQUIVALENCE (ARRAY(1),DETRNG(1))
 18             COMMON /JPARS2/
 19             COMMON /SPECS/
 20             REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
 21             COMMON /SPECS2/
 22             REAL*8 VEHS
       C
 23             REAL*8 ATTR
 24             DIMENSION CVALS(2,12),IVALS(12),VALS(12)
       C
       C  MUST BACKSPACE TO PCSITICN RECORD POINTER APPROPRIATE FOR READING
 25             BACKSPACE 2
       C
       C  GET TYPE OF ATTRIBUTE
 26             IFLD = IATTR2(ATTR,IRECTP)
 27             KFLD = IFLDPT(IRECTP) + IFLD
 28             ITYP = IFLDTP(KFLD)
       C
       C  GET COMPONENT IC AND COMPONENT TYPE
```

```
                    INPUT ROUTINES -    INATR2 SUBROUTINE
ISN
29              ID = INDCMP(ISPEC)
30              ICMP = IRECTP - 1
          C
          C   MARK SPECIFICATION AS PROCESSED
31              IRECS(JDONE, IREC) = 1
          C
          C   GET NUMBER OF FIELDS IN RECORD
32              NF = NFLDS(IRECTP)
          C   GET OFFSET OF RECORC FILE
33              IOFF = IFILPT(IRECTP)
          C
          C   GET POINTER TO RECORD HAVING THE APPROPRIATE ID TO STORE THE DATA
          C   IN FOR THIS COMPONENT; IF ONE DOESN'T EXIST, CREATE ONE.
34              IR = IRFIND(IRECTP,ID,ISPECV)
35              IF (IR .NE. NULL) GC TO 8
36                 IR = NEWREC(IRECTP)
37                 CALL STORE(IVECLE(ICMP,ISPECV),VFILE(IOFF),NF,JNEXT,IR)
38                 IVECLE(ICMP,ISPECV) = IR
          C   STORE THE IC
39                 CALL STORE(ID,VFILE(IOFF),NF,JID,IR)
          C
          C   GET THE COLUMN OF INTEREST IN THE EXISTING VEHICLE FILE
40        8     IV = IRECS(JVEH,IREC)
41              ICOL = ICOLS(IV)
          C
          C   BRANCH ACCORDING TO DATATYPE
42              DO 10 I=1,NTYPES
43                 IF (ITYP .EQ. ITYPES(I)) GO TO 15
44        10       CONTINUE
45              CALL ERR(10,'INATR2',RECNAM(IRECTP),ATTR,ITYP)
46              RETURN
          C
47        15    GO TO (20,40,60,80),I
          C
          C   READ & STORE A REAL VALUE
48        20    READ (2,100) (VALS(I),I=1,ICOL)
```

INPUT ROUTINES -   INATR2 SUBROUTINE

```
ISN
 49   100    FORMAT (12X,12(F8.0,2X))
      C  STORE VALUE IN THE RECCRD
 50          CALL STORE(VALS(ICOL),VFILE(IOFF),NF,IFLD,IR)
 51          RETURN
      C
      C  READ & STORE AN INTEGER VALUE
 52   40     READ (2,105) (IVALS(I),I=1,ICOL)
 53   105    FORMAT (12X,12(I8,2X))
 54          CALL STORE(IVALS(ICOL),VFILE(ICFF),NF,IFLD,IR)
 55          RETURN
      C
      C  READ & STORE AN ARRAY OF REAL VALUES
      C
      C  DETERMINE WHICH ARRAY FILE THE INFORMATION WILL BE STURED IN
 56   60     DO 62 IARR=1,NARRS
 57              IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND.
            +      IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 63
 58   62        CONTINUE
 59          CALL ERR(16,'INATR2',IRECTP,IFLD,ITYP)
      C
      C  NUMBER OF ELEMENTS IN ARRAY
 60   63     NV = IARDF2(JARSIZ,IARR)
      C  OFFSET IN FILE CONTAINING ARRAYS
 61          IOFFAR = IARDF2(JARRPT,IARR)
      C  GET A VACANT ARRAY CF TYPE IARR
 62          K = IARDF2(JARLST,IARR) + 1
 63          IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATR2',ATTR,
            +      IARDF2(JARNUM,IARR),IARR)
 64          IARDF2(JARLST,IARR) = K
      C  STOPE POINTER TC THIS ARRAY
 65          CALL STORE(K,VFILE(IOFF),NF,IFLD,IR)
      C  FOR EACH VALUE OF ARRAY SPECIFIED IN EXISTING VEHICLE FILE,
      C  READ, TRANSFORM, & STORE IT.
 66          DO 75 I=1,NV
 67             RFAD (2,100) (VALS(J),J=1,ICOL)
 68             CALL STORE(VALS(ICOL),ARRAY(IOFFAR),NV,I,K)
```

```
ISN
69   75        CONTINUE
70          RETURN
     C
     C  READ & STORE A CHARACTER VALUE (<= 8 CHARS)
71   80     READ (2,110) (CVALS(1,J),CVALS(2,J),J=1,ICOL)
72   110    FORMAT (12X,12(2A4,2X))
     C  DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
73          DO 82 IARR=1,NARRS
74             IF (IAPDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 83
75   82        CONTINUE
76          CALL ERR(16,'INATR2',IRECTP,IFLD,ITYP)
     C
     C  OFFSET IN FILE CONTAINING CHARACTER STRINGS
77   83     IOFFCH = IARDF2(JARRPT,IARR)
     C  GET A VACANT CHARACTER STRING RECORD
78          K = IARDF2(JARLST,IARR) + 1
79          IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATR2',ATTR,
          +       IARDF2(JARNUM,IARR),IARR)
80          IARDF2(JARLST,IARR) = K
     C  STORE POINTER TO THE CHARACTER RECORD
81          CALL STORE(K,VFILE(IOFF),NF,IFLD,IR)
     C  STORE CHARACTER DATA IN CHARACTER RECORD
82          CALL STORE(CVALS(1,ICOL),ARRAY(IOFFCH),2,1,K)
83          CALL STORE(CVALS(2,ICOL),ARRAY(IOFFCH),2,2,K)
84          RETURN
85          END
```

```
I SN
 1          SUBROUTINE INFUNC
      C
      C        SUBROUTINE READS A FORMATTED FILE OF DATA VARIABLE
      C  HEADERS FOLLOWED BY DATA FOR EACH VARIABLE. THE DATA VARIABLES
      C  CONSIST OF SEQUENTIAL VARIABLES IN A COMMON BLOCK.  EACH VARIABLE
      C  HEADER RECORD HAS THE FOLLOWING FIELDS:
      C          NAME     COL 1-8    VARIABLE NAME
      C         DIM 1        9-11    1ST DIMENSION OF VARIABLE IN PROGRAM
      C         DIM 2        12-14    2ND DIMENSION OF VARIABLE IN PROGRAM
      C         DIM 3        15-17    3RD DIMENSION OF VARIABLE IN PROGRAM
      C         STAR           19     '*' IF DATA FOR VARIABLE IS ALL ZEROS
      C      RED. DIM 1      20-22    (OPTIONAL) REDUCED DIMENSION APPLYING
      C                                 TO DATA
      C         REPL 1         23     (OPTIONAL) DATA GIVEN ALONG FIRST
      C                                   SECTION
      C                               OF DIMENSION IS TO BE REPLICATED ALONG
      C                                LATER SECTIONS IF FIELD CONTAINS "#"
      C         RED DIM 2     24-25    (OPTIONAL) REDUCED DIMENSION 2
      C           REPL 2        26     (OPTIONAL) REPLICATION SYMBOL (#)
      C        RED DIM 3     27-28     (OPTIONAL) REDUCED DIMENSION 3
      C          REPL 3       29       (OPTIONAL) REPLICATION FACTOR 3
      C         FTN FORMAT    31-78      FORTRAN FORMAT (IN PARENTHESES)
      C
      C     THE DATA VALUES FOR A VARIABLE FOLLOW THE HEADER CARD.  FOR
      C  ARRAYS THE NUMBER OF DATA VALUES EXPECTED IS THE PRODUCT OF THE
      C  NON-ZERO DIMENSIONS GIVEN ON THE HEADER CARD.  IF REDUCED DIMENSIONS
      C  ARE SPECIFIED THEY ARE USED INSTEAD OF THE ACTUAL DIMENSIONS IN
      C  COMPUTING THE NUMBER OF EXPECTED DATA VALUES.  FOR SCALARS, VARIABLES
      C  WITH NO DIMENSIONS, A SINGLE VALUE IS EXPECTED.  IF A '*' IS GIVEN
      C  IN THE STAR FIELD, ZERO DATA VALUES ARE EXPECTED & THE NEXT HEADER
      C  CARD SHOULD FOLLOW IMMEDIATELY.  THE FORMAT THAT THE DATA VALUES
      C  APPEAR IN IS UP TO THE USER.  THE FORMAT IS INDICATED ON THE HEADER
      C  CARD & MUST BE A VALID FORTRAN FORMAT SPECIFICATION.
      C     A BLOCK OF DATA VARIABLES MAY BE TERMINATED BY A RECORD WITH
      C  "ENDBLK" IN COLS 1-6.  ALL VARIABLES WITHIN A BLOCK MUST BE IN PROPER
```

INPUT ROUTINES - INFUNC SUBROUTINE

```
I SN
      C  SEQUENCE & IN 1:1 CORRESPONDENCE WITH THE VARIABLES OF THE ASSOCIATED
      C  COMMON BLOCK USED IN THE MAIN PROGRAM.  SEE THE USER'S MANUAL FOR A
      C  LIST OF THE INPUT VARIABLES EXPECTED, THEIR ORDER, & THEIR
      C  DIMENSIONS.
      C        THE DATA SET REFERENCE NUMBERS USED ARE AS FOLLOWS:
      C    1   INPUT
      C    8   OUTPUT   ECHO OF DATA FILE HEADER CARDS
      C
  2            COMMON /DATA/
  3            COMMON /DATA2/
  4            EQUIVALENCE (VEC(1),ACCEL(1))
      C
  5            REAL VEC(2),V(1000)
  6            REAL*8 VAR,ENDBLK,ENDFIL
  7            INTEGER FMT(12)
  8            DATA STAR1/1H*/,ENDBLK/6HENDBLK/,ENDFIL/6HENDFIL/
  9            DATA SHARP/1H#/
      C
 10            CALL FTNCMD('SET MODECHECK=OFF',17)
 11   10       JJ0 = 0
 12   11       READ (1,100,END=44) VAR,I,J,K,STAR,I2,S1,J2,S2,K2,S3,FMT
 13            WRITE (8,101) JJ0,VAR,I,J,K,STAR,I2,S1,J2,S2,K2,S3,FMT
 14   100      FORMAT (A8,3I3,1X,A1,I3,2(A1,I2),A1,1X,12A4)
 15   101      FORMAT (I6,2X,A8,3I3,1X,A1,I3,2(A1,I2),A1,1X,12A4)
 16            IF (VAR .EQ. ENDBLK .OR. VAR .EQ. ENDFIL) GO TO 44
 17            JSIZE = 1
 18            IF (I .NE. 0) JSIZE = JSIZE*I
 19            IF (J .NE. 0) JSIZE = JSIZE*J
 20            IF (K .NE. 0) JSIZE = JSIZE*K
 21            IF(JSIZE .GT. 5000) STOP 100
 22            JJ2 = JJ0 + JSIZE
 23            IF(JJ2 .GT. 5000) STOP 200
 24            JJ1 = JJ0 + 1
 25   14       IF (STAR .EQ. STAR1) GO TO 40
      C ***     IF ALL ZEROS, THEN SKIP THIS VARIABLE
 26            IF (I .NE. 0) GO TO 31
```

```
                    INPUT ROUTINES -   INFUNC SUBROUTINE
ISN
      C ***     IS THIS A SCALAR
 27          READ (1,FMT) VEC(JJ1)
 28          GO TO 40
      C
 29   31     IF (J .NE. 0) GO TO 32
      C ***    IS IT  A VECTOR ?
 30          IF (I2 .EQ. 0) I2 = I
 31          JJ3 = JJ0 + I2
 32          READ (1,FMT) (VEC(M),M=JJ1,JJ3)
 33          GO TO 40
      C
 34   32     IF (K .NE. 0) GO TO 36
      C ***     IS IT A 2-D ARRAY
 35          IF (I2 .NE. 0) GO TO 33
                READ (1,FMT) (VEC(M),M=JJ1,JJ2)
 37             GO TO 40
 38   33     IJ=I2*J2
 39          READ (1,FMT) (V(M),M=1,IJ)
 40          DO 34 I1=1,I2
 41             DO 34 J1=1,J2
 42                J11 = J1-1
 43                M1 = JJ0+I1+I*J11
 44                M2 = I1+I2*J11
 45   34           VEC(M1) = V(M2)
 46          GO TO 40
      C
 47   36     IF (I2 .NE. C) GC TO 38
 48          READ (1,FMT) (VEC(M),M=JJ1,JJ2)
 49          GO TO 40
 50   38     IJK = I2*J2*K2
 51          READ (1,FMT) (V(M),M=1,IJK)
 52          DO 39 I1=1,I2
 53             DO 39 J1=1,J2
 54                J11 = J1-1
 55                DO 39 K1=1,K2
 56                   K11 = K1-1
```

INPUT ROUTINES - INFUNC SUBROUTINE

```
ISN
57                     M1=JJ0+I1+I*J11+I*J*K11
58                     M2=I1+I2*J11+I2*J2*K11
59    39               VEC(M1) = V(M2)
      C
      C
60    40      IREP = 1
61            IF (S1 .EQ. SHARP) IREP = IREP*I
62            IF (S2 .EQ. SHARP) IREP = IREP*J
63            IF (S3 .EQ. SHARP) IREP = IREP*K
64            IF (IREP .EQ. 1) GO TO 43
65            J2SIZE = JSIZE/IREP
66            L1 = JJ0
67            DO 42 ICT=1,IREP
68                 L1 = L1 + J2SIZE
69                 DO 42 L2=1,J2SIZE
70    42              VEC(L1+L2)=VEC(JJ0+L2)
71    43      JJ0 = JJ0 + JSIZE
72            GO TO 11
73    44      RETURN
74            END
```

```
                    INPLT RCUTINES -   INIT SUBROUTINE
ISN
 1           SUBROUTINE INIT
        C    ROUTINE TO INITIALIZE VARIABLES AT BEGINNING OF RUN.
        C
 2           CCMMCN /VEH/
 3           DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
           +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +      IRELN(3,50),VFILE(1),IVFILE(1)
 4           EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
           +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
           +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
           +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
           +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
           +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5           EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6           COMMCN /VPPOTO/
 7           REAL*8 FLDNAM,RECNAM,DFAULT
 8           COMMON /NPARS/
 9           CCMMCN /OUTPUT/
10           LOGICAL*1 DF
11           REAL*8 XNAT,XMANUF,XMODEL
12           CCMMON /AUX/
13           REAL*8 BL8,DASHES
```

```
                    INPUT ROUTINES -   INIT SUBROUTINE
ISN
14            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
15            COMMON /ARRAYS/
16            REAL*8 MANUF,MODEL,NATICN
17            DIMENSICN ARRAY(1)
18            EQUIVALENCE (ARRAY(1),DETRNG(1))
19            COMMON /JPARS2/
20            CCMMON /SPECS/
21            REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
        C
        C   INITIALIZE IFLDPT BASED CN NUMBER OF FIELDS IN EACH TYPE OF
        C   RECORD
        C
22            N = 0
23            NN = 1
24            IFLDPT(1) = N
25            DO 10 I=1,NRECTP
26               N = N + NFLDS(I)
27               IFILPT(I) = NN
28               NN = NN + NFLCS(I) * NRECS(I)
29      10       IFLDPT(I+1) = N
        C
        C   DETERMINE NUMBER OF CELLS IN ALL RECORDS COMBINED
        C
30            NCELL = 0
31            DO 15 I=1,NRECTP
32      15       NCELL = NCELL + NFLDS(I) * NRECS(I)
        C
        C   INITIALIZE THE FIELCS CF ALL RECORDS TO NULL
        C
33            CALL NULIFY(VEHCLE,NCELL)
        C
        C   INITIALIZE VARIABLES THAT WILL CONTAIN TEXT TO BLANKS
        C
34            CALL INITVL(FLCNAM,NFLDRC*2,BLS)
35            CALL INITVL(RECNAM,NRECTP*2,BLS)
36            CALL INITVL(CMPCOD,NCMPTP,BLS)
```

```
ISN
37            CALL INITVL(UNITS,NFLDRC,BLS)
      C
      C  INITIALIZE COUNTERS TO ZERO
      C
38            LSTCMP = 0
39            LSTENG = 0
40            LSTPER = 0
41            CALL IZERO(LSTREC,NRECTP)
42            CALL IZERO(MAXNUM,NCMPTP)
43            CALL IZERO(MINNUM,NCMPTP)
44            CALL IZERO(NALTCS,NCMPTP)
      C
      C  INITIALIZE FIELD TYPES AND VALUES TO NULL
      C
45            CALL NULIFY(IFLDTP,NFLDRC)
46            CALL NULIFY(IFLDVL,NFLDRC)
47            CALL NULIFY(ICMPPT,NCMPTP)
      C
      C  INITIALIZE POINTERS TO ARRAYS
      C
48            NN = 1
49            DO 20 I=1,NARRS
50               IARDF2(JARLST,I) = 0
51               IARDF2(JARRPT,I) = NN
52               NA = IARDF2(JARSIZ,I) * IARDF2(JARNUM,I)
53    20         NN = NN + NA
      C
54            RETURN
55            END
```

```
                    INPUT ROUTINES -   INPROT SUBROUTINE
ISN
  1           SUBROUTINE INPROT
        C
        C       ROUTINE FOR REACING THE PROTOTYPE VEHICLE DESCRIPTION FILE AND
        C     STORING THIS INFORMATION IN INTERNAL RECORDS AND ARRAYS.
        C
  2           COMMON /VPROTO/
  3           REAL*8 FLDNAM,RECNAM,DFAULT
  4           COMMON /NPARS/
  5           COMMON /OUTPUT/
  6           LOGICAL*1 DF
  7           REAL*8 XNAT,XMANUF,XMODEL
  8           COMMON /AUX/
  9           REAL*8 BL8,DASHES
 10           EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 11           CCMMON /ARRAYS/
 12           REAL*8 MANUF,MODEL,NATICN
 13           DIMENSION ARRAY(1)
 14           EQUIVALENCE (ARRAY(1),DETRNG(1))
 15           COMMON /JPARS2/
 16           CCMMON /SPECS/
 17           REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
        C
 18           CCMMCN //
 19           DIMENSION ONAME(7)
 2)           EOUIVALENCE (CMPTYP,CNAME(1)), (CMPTYP,CMPTP2)
 21           REAL*8 V,VH,CMPTYP,RWNAME,PPARS,COMATR,COMEND,DFVAL
 22           DATA STAR/'*'/,VH/'VEHICLE'/
 23           DATA KRA/'RA'/,PPARS/'I. PERFO'/,COMATR/' A. COMM'/,
             +     CCMEND/' B. SPEC'/,KC8/'C8'/
        C
        C     INITIALIZATIONS
 24           LSTOUT = 0
 25           IARRAY = 0
 26           ICOM = 0
 27           KMP=0
```

```
                    INPUT ROUTINES -   INPROT SUBROUTINE
ISN
28              K = 10
29              LARDEF = 0
30              LEVEL = 0
      C
      C  READ VEHICLE HEADER; SKIP OVER COMMENT LINES
31    10        READ (3,100) C,V,VEHCAT
32    100       FORMAT (A1,A7,3X,A6)
33              IF (C .EQ. STAR) GO TO 10
34              IF (V .NE. VH) CALL ERR(1,'INPROT',V,0,0)
35              IREC = 1
36              IRECPT = IFLCPT(IREC)
37              RECNAM(IREC) = VH
38              ISTATE = 1
      C
      C  READ A NEW RECORD
39    15        READ(3,105,END=90) C,CMPTYP,FILL,CCOD, ITYPE
40    105       FORMAT(A1,A8,5A4,23X,A2)
      C
      C  IS THIS A COMMENT?  IF SO, SKIP INPUT DATA PROCESSING.
      C
41              IF(C.EQ.STAR) GO TO 30
      C
      C  IF THIS IS DATA, BACKSPACE AND READ INPUT RECORD AGAIN.
      C
42              BACKSPACE 3
      C
      C  READ DATA ON APPROPRIATE FORMAT AS INDICATED BY ITYPE;
      C  ITYPE.EQ.'C8' IMPLIES CHARACTER DFVAL, .NE.'C8' IMPLIES NUMERIC.
      C
43              IF(ITYPE.EQ.KC8) READ(3,107) CCOD,RWNAME,IFLDV,MIN,MAX,UNIT,
               +  DFVAL
44    107       FORMAT(25X,A4,A8,10X,I2,5X,2I3,2X,A4,2X,A8)
45              IF(ITYPE.NE.KC8) READ(3,109) CCOD,RWNAME,IFLDV,MIN,MAX,UNIT,
               +  DFVAL
46    109       FORMAT(25X,A4,A8,10X,I2,5X,2I3,2X,A4,2X,F8.0)
      C
```

INPUT ROUTINES -   INPROT SUBROUTINE

```
ISN
         C  CHECK FOR NEW CCMPONENT, INDICATED BY BLANK RWNAME AND IFLDV.
         C  IF TRUE, SKIP TC NEW COMPONENT SECTION.
         C
 47             IF (RWNAME .EQ. BL8 .AND. IFLDV .EQ. 0) GO TO 20
         C
         C  PROCESS NEW ATTRIBUTE OF CURRENT RECORD
 48             IF (IFLDV .GT. NFLDS(IREC)) CALL ERR(2,'INPROT',RWNAME,IFLDV,IREC)
 49             IFLDRC = IRECPT + IFLDV
 50             FLDNAM(IFLDRC) = RWNAME
 51             IFLDVL(IFLDRC) = IFLDV
 52             IFLDTP(IFLDRC) = ITYPE
 53             UNITS(IFLDRC) = UNIT
 54             DFAULT(IFLDRC) = DFVAL
 55             IF (ISTATE.NE.1) GO TO 18
 56             MINNUM(IFLDV) = MIN
 57             MAXNUM(IFLDV) = MAX
 58             KMP = KMP + 1
 59             KOUT(KMP) = IFLDV
 60      18     IF (ISTATE .EQ. 3 .OR. ICOM .EQ. 0) CALL  STONAM(DNAME,IFLDRC)
 61             IARRAY = 0
 62             IF (ITYPE .EC. KRA) IARRAY = 1
 63             GO TO 15
         C
         C  PROCESS A NEW CCMPONENT RECORD
 64      20     IPEC = IREC + 1
 65             ICMP = IREC - 1
 66             IF (IREC .GE. NRECTP) CALL ERR(3,'INPROT',IREC,NRECTP,0)
 67             IRECPT = IFLDPT(IREC)
 68             RECNAM(IREC) = CMPTYP
 69             CMPCOD(ICMP) = CCOD
 70             CALL STONAM(CNAME,0)
 71             GO TO 15
         C
 72      30     CONTINUE
         C
         C  COMMENT PROCESSING SECTION
```

```
                     INPUT ROUTINES -   INPROT SUBROUTINE
ISN
        C   END-OF-ARRAY
        C
 89            IARRAY = 0
        C
 90     38     CONTINUE
        C
        C   PROCESSING FOR CCMMENTS OTHER THAN ARRAY LABLES
        C
        C   IS THIS THE DEFINITICN OF AN ATTRIBUTE LEVEL ? IF SO SKIP IT
 91            IF (CMPTYP .EQ. BL8) GO TO 15
        C
        C   BRANCH TO CCDE ACCCRDING TO STATE
        C     STATE = 1: READING VEHICLE COMPONENT HEADERS; CHECK FOR
        C                  BEGINNING OF PERFORMANCE PARAMETERS
        C             2: READING PERFCRMANCE PARS; CHECK FOR END OF PERF PARS
        C             3: READING COMPONENTS & THEIR ATTRIBUTES; CHECK FOR
        C                  BEGINNING CF COMMON ATTRIBUTES
        C             4: READING CCMMCN ATTRIBUTES; CHECK WHETHER HAVE
        C                  REACHED END OF SECTION OR HAVE ALREADY READ THEM
        C
 92            GO TO (40,50,60,70),ISTATE
        C
        C   STATE 1.  IS THIS BEGINNING OF PERFORMANCE PARS ?
 93     40     IF (CMPTYP .NE. PPARS) GO TO 45
 94               IPPARS = LSTOUT + 1
 95               ISTATE = 2
 96               GO TO 15
        C   STORE NAME IN OUTNAM
 97     45     CALL STONAM(CNAME,0)
 98            GO TO 15
        C
        C   STATE 2.  IS THIS THE END OF THE PERFORMANCE PARS ?
 99     50     IF (CMPTYP .NE. DASHES) GO TO 55
100               ICMATR = LSTCUT + 1
101               ISTATE = 3
102               GO TO 15
```

```
ISN
      C
      C   CHECK WHETHER CURRENTLY PROCESSING ARRAY LABLES; IF NOT SKIP
      C   ARRAY LABEL SECTION
 73          IF (IARRAY .NE. 1) GO TO 38
      C
      C   ARRAY LABEL PROCESSING SECTION
      C
      C   CHECK FOR END OF CURRENT ARRAY LABLES; IF TRUE,
      C   SKIP TO END-OF-ARRAY
 74          IF (CMPTP2.NE.BLS.OR.FILL(1).EQ.BLS)GO TO 37
      C
      C   IF THIS IS LABEL OF ANOTHER LEVEL, STORE IT
 75   32     IF (CMPTYP .NE. BL8) GO TO 34
 76             LEVEL = LEVEL + 1
 77             IF (LEVEL .GT. NARLAB) CALL ERR(36,'INPROT',IREC,IFLDV,LEVEL)
 78             ILEVEL = ILEVEL + 1
 79             CALL COPY(ONAME(3),ARRLAB(1,LEVEL),5)
 80             GO TO 15
      C
      C   THIS IS A NEW DIMENSION NAME.  CHECK THAT PREVIOUS DIMENSION
      C   HAD RIGHT NUMBER OF LEVELS
 81   34     IF (LARDEF .EQ. 0) GO TO 36
 82             IF (ILEVEL .NE. IARDEF(JNLEVL,LARDEF)) CALL ERR(34,'INPROT',
           +      IREC,IFLDV,ILEVEL)
      C
      C   CHECK THAT CURRENT DIMENSION IS DEFINED IN IARDEF
 83   36     LARDEF = LARDEF + 1
 84          IF (IARDEF(JRECTP,LARDEF) .NE. IREC .OR. IARDEF(JFLDRC,LARDEF)
           +     .NE. IFLDV) CALL ERR(35,'INPROT',IREC,IFLDV,LARDEF)
 85          ILEVEL = 0
      C   STORE POINTER TO LEVEL
 86          IARDEF(JLEVPT,LARDEF) = LEVEL + 1
 87          GO TO 15
      C
 88   37     CONTINUE
      C
```

```
ISN
    C  SAVE NAME IN OUTNAM
103 55    CALL STONAM(CNAME,0)
104       GO TO 15
    C
    C  STATE 3.  IS THIS THE BEGINNING OF COMMON ATTRIBUTES ?
105 60    IF (CMPTYP .NE. COMATR) GO TO 65
106         ISTATE = 4
107         GO TO 15
    C
    C  HAS THE CCMMON ATTRIBUTE SECTION BEEN PROCESSED YET ?
108 65    IF (ICOM .NE. 0) GO TO 68
    C  IF NOT, STORE THESE FIRST FEW LINES IN A TEMPORARY LOCATION
    C  AND LATER MOVE THEM TO THE END OF THE COMMON ATTRIBUTE SECTION.
109         CALL COPY(ONAME,OUTNAM(1,NOUT-K),NOUTWD)
110         K = K - 1
111         GO TO 15
    C
    C  STORE NAME IN OUTNAM
112 68    IF (CMPTYP .EO. CASHES) GO TO 15
113       CALL STONAM(CNAME,0)
114       GO TO 15
    C
    C  STATE 4.  HAS ONE ALREADY READ THESE ATTRIBUTES ONCE ?
115 70     CONTINUE
    C  IS THIS THE END OF THE SECTION OF COMMON ATTRIBUTES ?
116       IF (CMPTYP .NE. COMEND .AND. CMPTYP .NE. DASHES) GO TO 75
117       ISTATE=3
118       IF(ICOM.NE.0) GO TO 15
    C
    C  AT THIS POINT ALL COMMENTS ASSOCIATED WITH COMMON ATTRIBUTES
    C  HAVE BEEN READ CNCE; WE SET POINTER AND FLAGS TO INDICATE THIS
    C
119         IEPARS = LSTOUT + 1
120       ICOM=1
    C  INSERT THE TEMPORARILY STORED NAMES INTO THE LIST AT THIS POINT
121         NR = 10 - K
```

```
 ISN
122              CALL COPY(OUTNAM(1,NOUT-10),OUTNAM(1,LSTOUT+1),NR*NOUTWD)
123              CALL IZERO(IOUTF(LSTOUT+1),NR)
124              LSTOUT = LSTOUT + NR
125              GO TO 15
       C  STORE NAME IN OUTNAM
126    75     IF(ICOM.EQ.0)CALL STONAM(ONAME,0)
127           GO TO 15
       C
128    90     RETURN
129           END
```

INPUT ROUTINES - INSPEC SUBROUTINE

```
ISN
 1          SUBROUTINE INSPEC
      C
      C     ROUTINE FOR READING AND INITIALLY STORING USER SEPCIFICATIONS
      C  IN INTERNAL FORM CLOSELY PARALLELING EXTERNAL FORM.
      C
 2          COMMON /NPARS/
 3          COMMON /AUX/
 4          REAL*8 BL8,DASHES
 5          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 6          COMMON /SPECS/
 7          REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
      C
 8          DIMENSION CMPHD(4)
 9          REAL*8 V
10          DATA VH/'VEHI'/,WT/'WEIG'/,GWT/'GROS'/,EVAL/'EVAL'/,
          +     DESIR/'DESI'/,CMPHD/'C','E','P','V'/,VEH/'VEH'/
      C
      C  READ & ECHO VEHICLE CLASS
11          READ (5,100) HEAD,V
12          WRITE (8,100) HEAD,V
13    100   FORMAT (A4,16X,A8)
14          IF (HEAD .NE. VH) CALL ERR(11,'INSPEC',HEAD,VH,0)
15          IF (V .NE. VEHCAT) CALL ERR(12,'INSPEC',V,VEHCAT,0)
      C
      C  READ & ECHO GROSS WEIGHT ESTIMATE FOR VEHICLE
16          READ (5,105) HEAD,VEHWT
17          WRITE (8,105) HEAD,VEHWT
18    105   FORMAT (A4,16X,F8.0)
19          IF (HEAD .NE. WT .AND. HEAD .NE. GWT) CALL ERR(11,'INSPEC',
          +     HEAD,WT,0)
      C
      C  READ & ECHO SET OF EVALUATION WEIGHTS FOR VARIOUS CATEGORIES
20          READ (5,110) HEAD,VEHEVL
21          WRITE (8,110) HEAD,VEHEVL
22    110   FORMAT (A4,16X,5F8.0)
```

```
                    INPUT ROUTINES -   INSPEC SUBROUTINE
ISN
23          IF (HEAD .NE. EVAL .AND. HEAD .NE. DESIR) CALL ERR(11,'INSPEC',
     +          HEAD,EVAL,0)
      C
      C  READ HEADER
24          READ (5,115,END=90) HEAD
25    115   FORMAT (A1)
      C
      C  BRANCH ACCORDING TO HEADER
26    5     DO 10 I=1,4
27             IF (CMPHD(I) .EQ. HEAD) GO TO (20,30,40,40),I
28    10       CONTINUE
29          CALL ERR(14,'INSPEC',HEAD,0,0)
      C
      C  READ & STORE COMPONENT SPECIFICATIONS
30    20    READ (5,115,END=90) HEAD
31          IF (HEAD .NE. BLS) GO TO 5
32          BACKSPACE 5
33          I = LSTCMP + 1
34          IF (I .GT. NCSPCS) CALL ERR(13,'INSPEC','CMPONENT',NCSPCS,0)
35          READ (5,120,END=90) HEAD,NUMCMP(I),TYPCMP(I),INDCMP(I),VEHCMP(I)
36          WRITE (8,120) HEAD,NUMCMP(I),TYPCMP(I),INDCMP(I),VEHCMP(I)
37    120   FORMAT (A1,I2,1X,A3,1X,I2,1X,A6)
38          LSTCMP = I
39          GO TO 20
      C
      C  READ & STORE ENGINEERING PARAMETER SPECIFICATIONS
40    30    I = LSTENG + 1
41          IF (I .GT. NESPCS) CALL ERR(13,'INSPEC','ENG PARS',NESPCS,0)
42          READ (5,125,END=90) HEAD,TYPENG(I),ATRENG(I),RELENG(I),VEHENG(I),
     +          MULENG(I),VL1ENG(I),ADDENG(I),VL2ENG(I),INDENG(I)
43          WRITE (8,125) HEAD,TYPENG(I),ATRENG(I),RELENG(I),VEHENG(I),
     +          MULENG(I),VL1ENG(I),ADDENG(I),VL2ENG(I),INDENG(I)
44    125   FORMAT (A1,A3,1X,A8,1X,A2,1X,A6,1X,A1,1X,F8.0,1X,A1,1X,F8.0,1X,I2)
45          IF (HEAD .NE. BLS) GO TO 5
46          LSTENG = I
47          GO TO 30
```

INPUT ROUTINES - INSPEC SUBROUTINE

```
ISN
        C
        C  READ & STORE PERFORMANCE PARAMETERS
  48    40     I = LSTPER + 1
  49           IF (I .GT. NPSPCS) CALL ERR(13,'INSPEC','PERF PAR',NPSPCS,0)
  50           READ (5,130,END=90) HEAD,TYPPER,ATRPER(I),RELPER(I),
              +     VEHPER(I),MULPER(I),VL1PER(I),ADDPER(I),VL2PER(I)
  51           WRITE (8,130) HEAD,TYPPER,ATRPER(I),RELPER(I),
              +     VEHPER(I),MULPER(I),VL1PER(I),ADDPER(I),VL2PER(I)
  52    130    FORMAT (A1,A3,1X,A8,1X,A2,1X,A6,1X,A1,1X,F8.0,1X,A1,1X,F8.0)
  53           IF (HEAD .NE. BLS) GO TO 5
  54           IF (TYPPER .NE. BLS .AND. TYPPER .NE. VEH) CALL ERR(15,'INSPEC',
              +     TYPPER,0,0)
  55           LSTPER = I
  56           GO TO 40
        C
  57    90     RETURN
  58           END
```

INPUT ROUTINES - INVEHC SUBROUTINE

```
ISN
 1          SUBROUTINE INVEHC
      C
      C        ROUTINE TO PRODUCE INTERNAL VEHICLE SPECIFICATIONS FROM THE USER'S
      C     INPUT SPECIFICATIONS AND A FILE OF EXISTING VEHICLES.
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +      IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6          COMMON /VPROTO/
 7          REAL*8 FLDNAM,RECNAM,DFAULT
 8          COMMON /JPARS/
 9          COMMON /JPARS1/
10          COMMON /NPARS/
11          COMMON /AUX/
```

```
ISN
12            REAL*8 BL8,DASHES
13            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
14            COMMON /ARRAYS/
15            REAL*8 MANUF,MODEL,NATION
16            DIMENSION ARRAY(1)
17            EQUIVALENCE (ARRAY(1),DETRNG(1))
18            COMMON /JPARS2/
19            CCMMCN /SPECS/
20            REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
21            CCMMON /SPECS2/
22            REAL*8 VEHS
        C
23            DIMENSION IALTS(12),VS(12)
24            REAL*8 ATTR,CMP,V,VS,VEHIC
25            DATA VEHIC/'VEHICLE'/,VEH/'VEH'/
        C
26            CALL NULIFY(ICMPS,NCMPTP)
27            CALL INITVL(VEHS,NVEHS*2,BLS)
28            CALL NULIFY(IRECS,NIRECI*NVR)
29            LSTVEH = 0
30            LVEH = 0
31            IVEHS = NULL
        C
        C  GO THROUGH INTERNAL FILE OF USER SPECIFICATIONS, NOTING THOSE THAT
        C  REFER TO AN EXISTING VEHICLE AND ARRANGING THEM SO THAT THE REQUIRED
        C  INFORMATION CAN BE EXTRACTED DURING ONE PASS THROUGH THE EXISTING
        C  VEHICLE FILE.  AT THE SAME TIME, PROCESS ALL SPECIFICATIONS THAT
        C  DON'T REQUIRE ACCESS TC THE EXISTING VEHICLE FILE.
        C
32            IF (LSTCMP .LE. 0) GO TO 50
        C  ITERATE OVER CCMPCNENT SPECIFICATICN RECORDS
33            DO 40 I=1,LSTCMP
        C  CHECK THAT COMPCNENT TYPE CODE IS LEGAL
34               IF (TYPCMP(I) .NE. VEH) GO TO 10
        C  THIS RECORD SPECIFIES A BASIC VEHICLE TO START WITH
35                  IF (VEHCMP(I) .EQ. BL8) CALL ERR(19,'INVEHC',I,
ISN
         +            0,0)
      C  GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
36                  K = ITABLE(VEHS,VEHCMP(I),NVEHS,2)
37                  IF (K .EQ. NULL) CALL ERR(20,'INVEHC',VEHCMP(I),NVEHS,0)
      C  FOR EACH COMPONENT OF VEFICLE,
      C  GET NEW RECORD TO STORE INFORMATICN TO GUIDE LATER FILE RETRIEVAL
      C  & STORE INFORMATION IN THIS RECORD
38                  DO 5 ICMP=1,NCMPTP
39      5              ICMPS(ICMP) = NEWVEH(ICMPS(ICMP),0,I,K)
40                  GO TO 40
      C
      C  GET COMPONENT INDEX
41      10          ICMP = LSERCH(CMPCOD,TYPCMP(I),NCMPTP,1)
42                  IF (ICMP .NE. NULL) GO TO 15
43                     CALL ERR(22,'INVEHC',TYPCMP(I),I,1)
44                     GO TO 40
      C  DOES THIS RECORC INCICATE THE NUMBER OF COMPONENTS OF A PARTICULAR
      C  TYPE DESIRED ?
45      15          IF (NUMCMP(I) .LE. 0) GO TO 20
      C  IF SO, ADJUST MIN & MAXNUM TO THIS NUMBER
46                  IF (NUMCMP(I) .LT. MINNUM(ICMP) .OR. NUMCMP(I) .GT.
         +             MAXNUM(ICMP)) CALL ERR(23,'INVEHC',NUMCMP(I),ICMP,I)
47                  MINNUM(ICMP) = NUMCMP(I)
48                  MAXNUM(ICMP) = NUMCMP(I)
      C
      C  IS AN EXISTING VEHICLE REFERRED TO ?
49      20          IF (VEHCMP(I) .EQ. BL8) GO TO 30
      C  GET INDEX OF VEFICLE IN VEHICLE NAME TABLE
50                  K = ITABLE(VEHS,VEHCMP(I),NVEHS,2)
51                  IF (K .EQ. NULL) CALL ERR(20,'INVEHC',VEHCMP(I),NVEHS,0)
      C  IF SO, GET NEW RECORD TO STCRE INFORMATION TO GUIDE LATER FILE
      C  RETRIEVAL & STORE INFO IN RECORD
52                  ICMPS(ICMP) = NEWVEH(ICMPS(ICMP),1,I,K)
53                  GO TO 40
      C
      C  COMPONENT SPEC SHOULD REFER TO AN ALTERNATIVE COMPONENT
```

INPUT ROUTINES - INVEHC SUBROUTINE

```
ISN
54    30          IF (INDCMP(I) .GE. 1 .AND. INDCMP(I) .LE. NALTCS(ICMP))
          +         GO TO 32
55                   CALL ERR(24,'INVEFC',INDCMP(I),TYPCMP(I),I)
56                   GO TO 40
      C  CREATE DUPLICATE RECORD OF THE DESIRED ALTERNATIVE COMPONENT
57    32             IRECTP = ICMP + 1
58                   IOFF = IFILPT(IRECTP)
59                   IR = NEWREC(IRECTP)
60                   CALL CCPY(VFILE(ICFF + (INDCMP(I)-1)*NFLDS(IRECTP)),
          +                VFILE(IOFF + (IR-1)*NFLDS(IRECTP)),NFLDS(IRECTP))
      C  SET NEXT FIELD CF NEW COMPONENT TO THE CURRENT LIST OF
      C  COMPONENTS
61                   CALL STORE(IVECLE(ICMP,ISPECV),VFILE(IOFF),NFLDS(IRECTP),
          +                JNEXT,IR)
      C  INDICATE COMPONENT TO BE SELECTED
62                IVECLE(ICMP,ISPECV) = IR
63    40          CONTINUE
      C
64    50       IF (LSTENG .LE. 0) GO TO 100
      C  ITERATE OVER ENGINEERING PARAMETER CONSTRAINTS SPECIFIED BY USER
65             DO 90 I=1,LSTENG
      C
      C  VERIFY THAT COMPONENT TYPE CODE SPECIFIED IS OK & GET CORRESPONDING
      C  COMPONENT INDEX
66                ICMP = LSERCH(CMPCOD,TYPENG(I),NCMPTP,1)
67                IF (ICMP .NE. NULL) GO TO 55
68                   CALL ERR(22,'INVEHC',TYPENG(I),I,2)
69                   GO TO 90
      C
      C  VERIFY THAT ATTRIBUTE IS SPECIFIED CORRECTLY; GET CORRESPONDING
      C  INDEX
70    55          IFLD = IATTR2(ATRENG(I),ICMP + 1)
71                IF (IFLD .NE. NULL) GO TO 60
72                   CALL ERR(25,'INVEHC',TYPENG(I),ATRENG(I),I)
73                   GO TO 90
      C
```

```
                    INPUT ROUTINES -   INVEHC SUBROUTINE
ISN
       C  VERIFY THAT RELCP IS APPROPRIATE
 74    60       DO 65 J=1,NRELCP
 75                IF (RELENG(I) .EQ. RELNAM(J)) GO TO 70
 76    65          CONTINUE
 77             CALL ERR(26,'INVEHC',TYPENG(I),ATRENG(I),RELENG(I))
 78             GO TO 90
       C
 79    70       KREL = IRELTP(J)
       C
       C  IS AN EXISTING VEHICLE REFERRED TO ?
 80             IF (VEHENG(I) .EQ. 8L8) GO TO 80
       C  GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
 81                K = ITABLE(VEHS,VEHENG(I),NVEHS,2)
 82                IF (K .EQ. NULL) CALL ERR(20,'INVEHC',VEHENG(I),NVEHS,0)
 83                IF (K .GT. LVEH) LVEH = K
       C  IF SO, GET NEW RECORD AND STORE INFO IN IT TO GUIDE LATER FILE
       C  RETRIEVAL
 84                ICMPS(ICMP) = NEWVEH(ICMPS(ICMP),2,I,K)
 85                GO TO 90
       C
       C  ENGINEERING PARAMETER SPEC SHOULD SPECIFY A VALUE.  STORE THIS
       C  AS A CONSTRAINT
 86    80       IR = NEWREL(ICMP+1,IFLD,KREL,VLIENG(I),INDENG(I))
 87    90       CONTINUE
       C
       C
       C  PROCESS PERFORMANCE PARAMETER SPECIFICATIONS
 88    100   IF (LSTPER .LE. 0) GO TO 150
       C  ITERATE OVER PERFORMANCE CONSTRAINTS SPECIFIED BY USER
 89          DO 140 I=1,LSTPER
       C
       C  VERIFY THAT ATTRIBUTE IS SPECIFIED CORRECTLY; GET CORRESPONDING INDEX
 90             IFLD = IATTR2(ATRPER(I),1)
 91             IF (IFLD .NE. NULL) GO TO 110
 92                CALL ERR(25,'INVEHC',VEH,ATRPER(I),0)
 93                GO TO 140
```

```
                    INPUT ROUTINES -    INVEHC SUBROUTINE
 ISN
      C
      C  VERIFY THAT RELOP IS APPROPRIATE
  94  110      DO 115 J=1,NRELOP
  95               IF (RELPER(I) .EQ. RELNAM(J)) GO TO 120
  96  115          CONTINUE
  97           CALL ERR(26,'INVEHC',VEH,ATRPER(I),RELPER(I))
  98           GO TO 140
      C
  99  120          KREL = IRELTP(J)
      C
      C  IS AN EXISTING VEHICLE REFERRED TO ?
 100           IF (VEHPER(I) .EQ. BL8) GO TO 130
      C  GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
 101               K = ITABLE(VEHS,VEHPER(I),NVEHS,2)
 102               IF (K .EQ. NULL) CALL ERR(20,'INVEHC',VEHPER(I),NVEHS,0)
      C  IF SO, GET NEW RECORD AND STORE INFO IN IT TO GUIDE LATER FILE
      C  RETRIEVAL
 103               IVEHS = NEWVEH(IVEHS,3,I,K)
 104               GO TO 140
      C
      C  PERFORMANCE PARAMETER SPEC SHOULD SPECIFY A VALUE.  STORE THIS AS
      C  A CONSTRAINT.
 105  130      IR = NEWREL(1,IFLD,KREL,VLIPER(I),0)
 106  140      CONTINUE
      C
      C
      C  READ & EXTRACT INFORMATION FROM EXISITING VEHICLE FILE
      C
 107  150   IF (VEHS(1) .EQ. BL8) GO TO 500
      C
      C  READ VEHICLE CATEGORY HEADER AND CHECK AGAINST EXPECTATION
      C
 108        READ (2,900,END=500) V
 109  900   FORMAT (A8)
 110        IF (V .NE. VEHCAT) CALL ERR(18,'INVEHC',V,VEHCAT,0)
      C
```

```
                 INPUT ROUTINES -    INVEHC SUBROUTINE

C  DETERMINE THE LAST COMFCNENT THAT NEED TO READ INFORMATION ABOUT
C
           LCMP = 0
           DO 155 ICMP=1,NCMPTP
              IF (ICMPS(ICMP) .NE. NULL) LCMP = ICMP
155           CONTINUE
C
C  READ VEHICLE RECORD
           READ (2,910,END=380) V,VS
910        FORMAT (A8,4X,12(A8,2X))
C
C  IDENTIFY THE VEFICLES WHCIH WANT TO EXTRACT INFORMATION ABOUT
           DO 180 IV=1,LVEH
              DO 165 ICCL=1,NCMPZL
                 IF (VS(ICOL) .EQ. VEHS(IV)) GO TO 170
165              CONTINUE
              GO TO 180
C
C  FOUND A VEHICLE OF INTEREST; MARK IT FOUND & REMEMBER ITS COLUMN
170           ICOLS(IV) = ICOL
180           CCNTINUE
C
C  READ VEHICLE COMPONENT POINTERS
190        READ (2,920) CMP,IALTS
920        FORMAT (2X,A8,12(8X,I2))
           IF (CMP .EQ. BL8 .OR. CMP .EQ. DASHES) GO TO 220
C
C  IDENTIFY WHETHER THIS IS A COMPONENT OF INTEREST
           DO 200 ICMP=1,NCMPTP
              IF (ICMPS(ICMP) .NE. NULL .AND. RECNAM(ICMP+1) .EQ. CMP)
     +           GO TO 205
200           CONTINUE
           GO TO 190
C
C  COMPONENT IS OF INTEREST; REMEMBER WHICH COLUMN
C  EACH COMPONENT CF INTEREST OF THIS TYPE IS IN.
```

```
                         INPUT ROUTINES -   INVEHC SUBROUTINE
ISN
      C  ITERATE OVER CONSTRAINTS INVOLVING THIS COMPONENT & STORE COLUMN
      C  INFORMATICN
 131  2)5    IREC = ICMPS(ICNP)
 132  208    IF (IREC .EQ. NULL) GO TO 190
               IV = IRECS(JVEH,IREC)
 133
 134           ICOL = ICOLS(IV)
 135           IF (IRECS(JCOL,IREC) .EQ. NULL) IRECS(JCOL,IREC) = IALTS(ICOL)
 136           IREC = IRECS(JNXT,IREC)
 137           GO TO 208
      C
      C  READ VEHICLE PERFORMANCE PARAMETERS
 138  22)    READ (2,93)) ATTR
 139  930    FORMAT (2X,A8)
 140         IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 24)
      C  IS THIS ATTRIBUTE OF INTEREST ?
 141         IREC = IVEHS
 142  225    IF (IREC .EQ. NULL) GO TO 22)
 143           ISPEC = IRECS(JSPEC,IREC)
 144           IF (ATRPER(ISPEC) .NE. ATTR) GO TO 230
      C  THIS IS ATTRIBUTE OF INTEREST; READ IT
 145           CALL INATTR(ATTR,IREC,ISPEC,1)
 146  230      IREC = IRECS(JNXT,IREC)
 147           GO TO 225
      C
      C  READ VEHICLE COMPONENT HEADER
 148  24)    READ (2,94),END=3))) CMP,IALTS
 149  940    FORMAT (A8,2X,12(8X,I2))
      C  DETERMINE COMPONENT TYPE
 15)         DO 250 ICMP=1,NCMPTP
 151           IF (RECNAM(ICMP+1) .EQ. CMP) GO TO 255
 152  25)      CONTINUE
 153         CALL ERR(28,'INVEHC',CMP,0,0)
      C
      C  IS THIS A COMPONENT OF INTEREST ?
 154  255    IF (ICMP .GT. LCMP) GO TO 300
 155         IF (ICMPS(ICNP) .NE. NULL) GO TO 260
```

```
C
C  IF NOT, SKIP OVER ITS ATTRIBUTE RECORDS
258   READ (2,930,END=300) ATTR
      IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 240
      GO TO 258
C
C  ARE INTERESTED IN THIS COMPONENT.  ITERATE OVER USER
C  SPECIFICATIONS FOR EACH ATTRIBUTE OF THE COMPONENT TO SEE WHETHER
C  THE ATTRIBUTE IS OF INTEREST.  FIRST READ THE NEXT ATTRIBUTE.
C
260   IRECTP = ICMP + 1
262   READ (2,930,END=300) ATTR
      IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 240
C
C  IS THIS AN ATTRIBUTE OF INTEREST ?
      IREC = ICMPS(ICMP)
265   IF (IREC .EQ. NULL) GC TO 262
         ISPEC = IRECS(JSPEC,IREC)
         ITYP = IRECS(JTYP,IREC)
         IF (ITYP .EQ. 2) GC TO 270
C  SPECIFICATION CALLS FOR THIS ATTRIBUTE TO BE USED AND STORED
C  AS A CONSTANT
         CALL INATR2(ATTR,IREC,ISPEC,IRECTP)
         GO TO 275
C  CHECK WHETHER ATTRIBUTE IS OF INTEREST
270      IF (ATTR .NE. ATRENG(ISPEC)) GO TO 275
C  SPECIFICATICN CALLS FOR THIS ATTRIBUTE TO BE READ AND STORED AS
C  A CONSTRAINT
         CALL INATTR(ATTR,IREC,ISPEC,IRECTP)
C  GET NEXT SPECIFICATION RELEVANT TO THIS COMPONENT
275      IREC = IRECS(JNXT,IREC)
         GO TO 265
C
C  CHECK THAT ALL USER SPECIFICATIONS REFERRING TO THE
C  EXISTING VEHICLE FILE HAVE BEEN PROCESSED.
300   IPREV = NULL
```

INPUT ROUTINES - INVEHC SUBROUTINE

```
 ISN
174           DO 310 IV=1,LSTVEH
175              IF (IRECS(JDONE,IV) .NE. NULL) GO TO 310
176                 IRECS(JDONE,IV) = IPREV
177                 IPREV = IV
178     310      CONTINUE
179           IF (IPREV .NE. NULL) CALL ERR(32,'INVEHC',IPREV,0,0)
180           RETURN
        C
181     380   CALL ERR(33,'INVEHC',LSTVEH,0,0)
        C
182     500   RETURN
183           END
```

C.4: SOLUTION ROUTINES

This section contains listings of the solution routines including the main "backtracking" algorithm (GENVEH), the function OKCOMP used to check compatibility of components with user specifications, the function COMPAT, used to check component compatibility and the routines used to estimate vehicle engineering and performance parameters (DIMENS, FPOWER, MOBILE, and PAIT). See section C-2 for listings of the labeled COMMON blocks referred to in these routines.

SOLUTION ROUTINES - FUNCTION COMPAT

```
ISN
 1            LOGICAL FUNCTION COMPAT(ICMP,IR)
      C
      C          ROUTINE TO TEST WHETHER A TRIAL COMPONENT OF A GIVEN TYPE IS
      C       COMPATIBLE WITH PREVIOUS COMPONENTS SELECTED.  RETURNS .TRUE. IF SO,
      C       .FALSE. OTHERWISE.
      C
      C       INPUT PARAMETERS:
      C          ICMP    COMPONENT TYPE
      C          IR      COMPONENT
      C
 2            COMMON /VEH/
 3            DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
           +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +      IRELN(3,50),VFILE(1),IVFILE(1)
 4            EQUIVALENCE  (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
           +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
           +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
           +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
           +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
           +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5            EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6            COMMON /VPROTO/
```

SOLUTION ROUTINES - FUNCTION COMPAT

```
ISN
  7             REAL*8 FLDNAM,RECNAM,DFAULT
  8             COMMON /JPARS/
  9             COMMON /JPARS1/
 10             COMMON /NPARS/
 11             COMMON /AUX/
 12             REAL*8 BL8,DASHES
 13             EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 14             COMMON /SPECS/
 15             REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
      C
 16             COMPAT = .TRUE.
 17             GO TO (90,90,10,90,90,  90,90,90,90,90,  20,90,90,30,90,
               +      90,90,90,90,90,  90,90,90,90,90,  90,90,90,90,90,  90),ICMP
      C
      C  CHECK THAT MAIN GUN IS NOT TOO LARGE FOR WEIGHT OF VEHICLE
 18   10        WTMIN = FMAING(JMINWT,IR)
 19             IF (WTMIN .EQ. FNULL) GO TO 90
 20             IF (WTMIN .GT. VEHWT) COMPAT = .FALSE.
 21             GO TO 90
      C
      C  CHECK THAT ORDNANCE HAS SAME CALIBER AS AT LEAST ONE OF THE GUNS
      C  SELECTED
 22   20        CAL = DVAL2(AMMO(JCALIB,IR),ICMP,JCALIB,'MM  ')
 23             DO 26 IC=JMAING,JADGUN
 24   22           IGR = IVECLE(IC,ICANDV)
 25                IF (IGR .EQ. NULL) GO TO 26
 26                IRECTP = IC + 1
 27                IOFF = IFILPT(IRECTP)
 28                GCAL = DVAL(VFILE(IOFF),NFLDS(IRECTP),JCALIB,IGR,IRECTP,'MM  ',D)
 29                IF (GCAL-CAL .LT. .001 .AND. CAL-GCAL .LT. .001) GO TO 90
 30                IGR = IDVAL(VFILE(IOFF),NFLDS(IRECTP),JNEXT,IGR,IRECTP,'-   ',D)
 31                GO TO 22
 32   26           CONTINUE
 33             COMPAT = .FALSE.
 34             GO TO 90
      C
```

SOLUTION ROUTINES - FUNCTION COMPAT

```
ISN
     C  CHECK THAT TRANSMISSION IS COMPATIBLE WITH ENGINE
35   30    KEYE = IDVAL(ENGINE,NFLDS(JENGIN+1),JTRNRQ,IVECLE(JENGIN,ICANDV),
          +     JENGIN+1,'-',DF)
36         KEYT = IDVAL2(TRANSM(JKEY,IR),JTRANS,JKEY,'-    ')
37         IF (KEYE .NE. KEYT) COMPAT = .FALSE.
38         GO TO 90
     C
39   90    RETURN
40         END
```

SOLUTION ROUTINES - DIMENS SUBROUTINE

```
ISN
 1             SUBROUTINE DIMENS(IV,IRETN)
        C
        C      ROUTINE TO CALCULATE THE DIMENSIONS OF A TANK BASED ON AN ASSUMED
        C   GEOMETRICAL CONFIGURATION (SEE ARTICLE BY R. LAWSON IN THE TANK
        C   WEAPON SYSTEM, SYSTEM RESEARCH GROUP, OSU, SEPT. 1968)
        C      ALSO CALCULATES SELECTED MOBILITY CHARACTERISTICS.
        C
        C   INPUT PARAMETERS:
        C      IV      SPECIFIC VEHICLE OF INTEREST
        C
        C   OUTPUT PARAMTERS:
        C      IRETN   COMPONENT TYPE TO TRY NEW SELECTION IF BACKUP IS NEEDED;
        C   SET TO ZERO OTHERWISE.
        C
 2             COMMON /VEH/
 3             DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
             +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
             +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
             +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
             +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
             +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
             +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
             +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
             +      IRELN(3,50),VFILE(1),IVFILE(1)
 4             EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
             +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
             +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
             +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
             +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
             +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
             +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
             +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
             +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
             +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
             +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
                   SOLUTION ROUTINES - DIMENS SUBROUTINE
ISN
           +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
           +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
   5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
   6          COMMON /VPROTO/
   7          REAL*8 FLDNAM,RECNAM,DFAULT
   8          COMMON /JPARS/
   9          COMMON /JPARS1/
  10          COMMON /NPARS/
  11          COMMON /AUX/
  12          REAL*8 BL8,DASHES
  13          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
  14          COMMON /ARRAYS/
  15          REAL*8 MANUF,MODEL,NATION
  16          DIMENSION ARRAY(1)
  17          EQUIVALENCE (ARRAY(1),DETRNG(1))
  18          COMMON /JPARS2/
  19          COMMON /DATA/
  20          COMMON /DATA2/
        C
  21          DATA PI/3.14159/
        C
        C   GET POINTERS TO COMPONENTS OF VEHICLE
  22          LTURET = IVECLE(JTURET,IV)
  23          LHULL = IVECLE(JHULL,IV)
  24          LMAING = IVECLE(JMAING,IV)
  25          LENGIN = IVECLE(JENGIN,IV)
  26          LTRANS = IVECLE(JTRANS,IV)
  27          LFINDR = IVECLE(JFINDR,IV)
  28          LROADW = IVECLE(JROADW,IV)
  29          LTRACK = IVECLE(JTRACK,IV)
  30          LSKIRT = IVECLE(JSKIRT,IV)
  31          LAMMO = IVECLE(JAMMO,IV)
        C
        C   TURRET DIAMETER
  32          X1 = DVAL2(MAINGN(JX1,LMAING),JMAING,JX1,'IN  ')
  33          X2 = DVAL2(TURRET(JX2,LTURET),JTURET,JX2,'IN  ')
```

```
ISN
34          XTP = 2. * (X1+X2)
35          TURRET(JXTP,LTURET) = XTP
      C
      C  HULL INSIDE WIDTH & VEHICLE WIDTH
36          X5 = DVAL2(ENGINE(JX5,LENGIN),JENGIN,JX5,'IN  ')
37          X6 = DVAL2(ENGINE(JX6,LENGIN),JENGIN,JX6,'IN  ')
38          X7 = DVAL2(TRANSM(JX7,LTRANS),JTRANS,JX7,'IN  ')
39          X8 = DVAL2(FINLDR(JX8,LFINDR),JFINDR,JX8,'IN  ')
40          XHPT = AMAX1(X5+2.*X6,X7+2.*X8)
41          XH = AMAX1(TURRET(JXTP,LTURET),XHPT)
42          HULL(JXH,LHULL) = XH
43          X3 = DVAL2(HULL(JX3,LHULL),JHULL,JX3,'IN  ')
44          X4 = DVAL2(TRACK(JX4,LTRACK),JTRACK,JX4,'IN  ')
45          X0 = 2.*X3 + 2.*X4 + XH
46          VEHCLE(JVWID,IV) = X0
      C
      C  LENGTH OF HULL
47          THFU = DVAL2(HULL(JTHFU,LHULL),JHULL,JTHFU,'IN  ')
48          THFL = DVAL2(HULL(JTHFL,LHULL),JHULL,JTHFL,'IN  ')
49          GAMMAU = DVAL2(HULL(JGAMU,LHULL),JHULL,JGAMU,'RAD ')
50          COSGU = COS(GAMMAU)
51          TANGU = TAN(GAMMAU)
52          Y1 = THFU / COSGU
53          Y2 = DVAL2(HULL(JY2,LHULL),JHULL,JY2,'IN  ')
54          Y3 = DVAL2(HULL(JY3,LHULL),JHULL,JY3,'IN  ')
55          K1 = DVAL2(TURRET(JK1,LTURET),JTURET,JK1,'IN  ')
56          XTP = DVAL2(TURRET(JXTP,LTURET),JTURET,JXTP,'IN  ')
57          Y4 = XTP + K1
58          Y5 = DVAL2(HULL(JY5,LHULL),JHULL,JY5,'IN  ')
59          Y6 = DVAL2(ENGINE(JY6,LENGIN),JENGIN,JY6,'IN  ')
60          Y7 = DVAL2(TRANSM(JY7,LTRANS),JTRANS,JY7,'IN  ')
61          Y8 = DVAL2(TRANSM(JY8,LTRANS),JTRANS,JY8,'IN  ')
62          Y9 = DVAL2(HULL(JY9,LHULL),JHULL,JY9,'IN  ')
63          YH = Y2 + Y3 + Y4 + Y5 + Y6 + Y7 + Y8
64          HULL(JYH,LHULL) = YH
65          Y0 = Y1 + YH + Y9
```

SOLUTION ROUTINES - DIMENS SUBROUTINE

```
ISN
66           VEHCLE(JVLEN,IV) = YO
       C
       C  TRACK GROUND CONTACT LENGTH
67           D1 = DVAL2(ROADWH(JD1,LROADW),JROADW,JD1,'IN  ')
68           D2 = DVAL2(ROADWH(JD2,LROADW),JROADW,JD2,'IN  ')
69           D3 = DVAL2(ROADWH(JD3,LROADW),JROADW,JD3,'IN  ')
70           T = DVAL2(TRACK(JT,LTRACK),JTRACK,JT,'IN  ')
71           Z1 = DVAL2(ROADWH(JZ1,LROADW),JROADW,JZ1,'IN  ')
72           Z14 = DVAL2(ROADWH(JZ14,LROADW),JROADW,JZ14,'IN  ')
73           Z15 = DVAL2(ROADWH(JZ15,LROADW),JROADW,JZ15,'IN  ')
74           BF = DVAL2(ROADWH(JBETAF,LROADW),JROADW,JBETAF,'RAD ')
75           BR = DVAL2(ROADWH(JBETAR,LROADW),JROADW,JBETAR,'RAD ')
76           SINBF = SIN(BF)
77           TANBF = TAN(BF)
78           COSBF = SINBF / TANBF
79           SINBR = SIN(BR)
80           TANBR = TAN(BR)
81           COSBR = SINBR / TANBR
82           TD3Z1 = T + D3/2. - Z1
83           ZZ14 = (SQRT(((D1+D3)/2.)**2 - ((D1-D3)/2.)**2)
          +       + (D1-D3) / (2.*TANBF)) * SINBF + TD3Z1
84           ZZ15 = (SQRT(((D2+D3)/2.)**2 - ((D2-D3)/2.)**2)
          +       + (D2-D3) / (2.*TANBR)) * SINBR + TD3Z1
85           IF (Z14 .GE. ZZ14) GO TO 5
86              CALL ERR(40,'DIMENS','Z14','IDLER',Z14)
87              Z14 = ZZ14
88     5     IF (Z15 .GE. ZZ15) GO TO 10
89              CALL ERR(40,'DIMENS','Z15','SPROCKET',Z15)
90              Z15 = ZZ15
91     10    C1 = D1/2. + T
92           C2 = D2/2. + T
93           Z1M = Z1 - (T + D3/2.)
94           YGC = YO - (C1 + C2 +
          +       (2.*COSBF*(Z14+Z1M) - (D1-D3)) / (2.*SINBF) +
          +       (2.*COSBR*(Z15+Z1M) - (D1-D3)) / (2.*SINBR))
95           VEHCLE(JYGC,IV) = YGC / 12.
```

```
                   SOLUTION ROUTINES - DIMENS SUBROUTINE
 ISN
        C
        C   HEIGHTS
  96            Z2 = DVAL2(HULL(JZ2,LHULL),JHULL,JZ2,'IN  ')
  97            Z3 = DVAL2(HULL(JZ3,LHULL),JHULL,JZ3,'IN  ')
  98            Z4 = DVAL2(HULL(JZ4,LHULL),JHULL,JZ4,'IN  ')
  99            Z5 = DVAL2(TURRET(JZ5,LTURET),JTURET,JZ5,'IN  ')
 100            Z10 = DVAL2(ENGINE(JZ10,LENGIN),JENGIN,JZ10,'IN  ')
 101            Z11 = DVAL2(ENGINE(JZ11,LENGIN),JENGIN,JZ11,'IN  ')
 102            Z12 = DVAL2(HULL(JZ12,LHULL),JHULL,JZ12,'IN  ')
 103            Z13 = DVAL2(TURRET(JZ13,LTURET),JTURET,JZ13,'IN  ')
        C
 104            ALPHAF = DVAL2(MAINGN(JDPRES,LMAING),JMAING,JDPRES,'RAD ')
 105            B = DVAL2(MAINGN(JODIAM,LMAING),JMAING,JODIAM,'IN  ')
 106            Y22 = Y1 + Y2 + Y3 + Y4/2. - (Z1+Z2+Z4)*TANGU/2.
 107            Y23 = DVAL2(TURRET(JY23,LTURET),JTURET,JY23,'IN  ')
 108            Y24 = Y4/2. + Y5 + Y6 + Y7 + Y8
 109            SINAF = SIN(ALPHAF)
 110            B2SAF = B/(2. * SINAF)
 111            TANAF = TAN(ALPHAF)
 112            COSAF = SINAF / TANAF
 113            X02 = (X0/2.) ** 2
 114            ZZ5 = (SQRT(Y22**2 + X02) + B2SAF - Y23)*TANAF
 115            IF (TURRET(JZ5,LTURET) .EQ. FNULL) GO TO 12
 116            IF (Z5 .GE. ZZ5) GO TO 15
 117               CALL WARN(2,'DIMENS','FRONT',ZZ5,Z5)
 118    12         Z5 = ZZ5
 119               TURRET(JZ5,LTURET) = Z5
 120    15      ZZ13 = (SQRT(Y24**2 + X02) + B2SAF - Y23)*TANAF
 121            IF (TURRET(JZ13,LTURET) .EQ. FNULL) GO TO 18
 122            IF (Z13 .GE. ZZ13) GO TO 20
 123               CALL WARN(2,'DIMENS','REAR',ZZ13,Z13)
 124    18         Z13 = ZZ13
 125               TURRET(JZ13,LTURET) = Z13
 126    20      Y20 = DVAL2(MAINGN(JY20,LMAING),JMAING,JY20,'IN  ')
 127            Y21 = DVAL2(MAINGN(JY21,LMAING),JMAING,JY21,'IN  ')
 128            Z6 = (Y20+Y21)*SINAF + B/(2.*COSAF)
```

SOLUTION ROUTINES - DIMENS SUBROUTINE

```
 ISN
       C
 129          Z7 = DVAL2(TURRET(JZ7,LTURET),JTURET,JZ7,'IN  ')
 130          Z8 = DVAL2(HULL(JZ8,LHULL),JHULL,JZ8,'IN  ')
 131          Z9 = DVAL2(HULL(JZ9,LHULL),JHULL,JZ9,'IN  ')
 132          ZMAX2 = AMAX1(Z3+Z4+Z5+Z6,Z10+Z11+Z12+Z13+Z6,Z8+Z9)
 133          Z0 = Z1 + Z2 + ZMAX2 + Z7
 134          VEHCLE(JVHT,IV) = Z0
       C
       C  AREA & WEIGHT OF HULL
 135          ZH = AMAX1(Z3,Z10+Z11)
 136          HULL(JZH,LHULL) = ZH
 137          IARMHL = IHULL(JARMTP,LHULL)
 138          D = DENSTY(IARMHL)
 139          AHUF = XH * ZH/(2. * COSGU)
 140          WHUF = AHUF * THFU * D
       C
 141          AHFD = XH * (Y2 + Y3 + Y4/2. - ZH*TANGU/2.) - PI*Y4**2/8.
 142          WHFD = AHFD * Z4 * D
       C
 143          AHRD = XH*(Y4/2. + Y5 + Y6 + Y7) - PI*Y4**2/8.
 144          WHRD = AHRD * Z12 * D
       C
 145          AHUB = XH * ZH / 2.
 146          WHUB = AHUB * Y9 * D
       C
 147          DELTAD = DVAI2(HULL(JDELD,LHULL),JHULL,JDELD,'RAD ')
 148          COSDD = COS(DELTAD)
 149          TANDD = TAN(DELTAD)
 150          AHLB = XH * ZH/(2. * COSDD)
 151          WHLB = AHLB * Y9 * D
       C
 152          GAMMAD = DVAL2(HULL(JGAMD,LHULL),JHULL,JGAMD,'RAD ')
 153          TANGD = TAN(GAMMAD)
 154          COSGD = COS(GAMMAD)
 155          AHB = XH * (YH - ZH*TANDD/2. - ZH*TANGD/2.)
 156          WHB = AHB * Z2 * D
 ISN
        C
 157            AHS = 2. * YH * ZH - (ZH**2/2.) * (TANDD + TANGD + TANGU)
 158            WHS = AHS * X3 * D
        C
 159            AHLF = XH * ZH / (2. * COSGD)
 160            WHLF = AHLF * THFL * D
        C
 161            WH = WHUF + WHFD + WHRD + WHUB + WHLB + WHB + WHS + WHLF
 162            HULL(JWT,LHULL) = WH
        C
        C  WEIGHT OF TURRET ARMOR
 163            AI = .717 * Y4
 164            TURRET(JAI,LTURET) = 2. * AI
 165            CI = .59 * Y4
 166            BI = Z0 - (Z1 + Z2 + Z3 + Z4 + Z7)
 167            TURRET(JBI,LTURET) = 2. * BI
 168            IARMTU = IDVAL2(TURRET(JARMTP,LTURET),JTURET,JARMTP,'-    ')
 169            DT = DENSTY(IARMTU)
 170            TTF = DVAL2(TURRET(JTTF,LTURET),JTURET,JTTF,'IN  ')
 171            TTS = DVAL2(TURRET(JTTS,LTURET),JTURET,JTTS,'IN  ')
 172            TTU = DVAL2(TURRET(JTTU,LTURET),JTURET,JTTU,'IN  ')
 173            TTB = DVAL2(TURRET(JTTB,LTURET),JTURET,JTTB,'IN  ')
 174            WT = .6667*PI*((AI + TTF/2. + TTB/2.) * (BI + Z7) * (CI + TTS)
              +      - (AI*BI*CI))*DT + (4.*PI*AI*CI - PI*Y4**2/4.)*TTU*DT
 175            TURRET(JWT,LTURET) = WT
        C
        C  COMPUTE WEIGHT OF TRACK SKIRTS, IF ANY
 176            IF (LSKIRT .EQ. NULL) GO TO 30
 177               IARMTP = IDVAL2(SKIRT(JTYPE,LSKIRT),JSKIRT,JTYPE,'-    ')
 178               DS = DENSTY(IARMTP)
 179               HS = DVAL2(SKIRT(JSKHT,LSKIRT),JSKIRT,JSKHT,'IN  ')
 180               TS = DVAL2(SKIRT(JSKTHK,LSKIRT),JSKIRT,JSKTHK,'IN  ')
 181               WS = 2.*DS*TS*(Y0*HS + Y0*X4 + 2.*X4*HS)
 182               SKIRT(JWT,LSKIRT) = WS
        C
        C
```

```
                    SOLUTION ROUTINES - DIMENS SUBROUTINE
 ISN
      C  SUM WEIGHT OF ALL COMPONENTS
 183  30    W = SUMCMP(IV,JWT,'LB  ')
 184        WTON = W / 2000.
 185        VEHCLE(JVWT,IV) = W
      C
      C  SUM INTERNAL VOLUMES OF COMPONENTS
 186        VSUM = SUMCMP(IV,JVOL,'FT3 ')
      C
      C  CALCULATE INTERNAL VOLUME OF THE VEHICLE ENVELOPE
 187        VH = AHS * XH / 2.
 188        HULL(JVOL,LHULL) = VH / 1728.
 189        VT = .6667 * PI * AI * BI * CI
 190        TURRET(JVOL,LTURET) = VT / 1728.
 191        VOLENV = VH + VT
 192        VOLFT3 = VOLENV / 1728.
 193        VEHCLE(JVVOL,IV) = VOLFT3
      C
      C
      C  MOBILITY PARAMETERS
      C
      C  TREAD & L/T RATIO
 194        TREAD = VEHCLE(JVWID,IV) - X4
 195        RLT = YGC / TREAD
 196        VEHCLE(JLT,IV) = RLT
      C
      C  GROSS HP / TON
 197        HP = DVAL2(ENGINE(JHP,LENGIN),JENGIN,JHP,'HP  ')
 198        HPTON = HP / WTON
 199        VEHCLE(JGHPTN,IV) = HPTON
      C
      C  SPROCKET HP / TON
 200        EFFTR = DVAL2(TRANSM(JEFFIC,LTRANS),JTRANS,JEFFIC,'-    ')
 201        EFFFD = DVAL2(FINLDR(JEFFIC,LFINDR),JFINDR,JEFFIC,'-    ')
 202        SHPTON = HP * (EFFTR/100.) * (EFFFD/100.) / WTON
 203        VEHCLE(JSHPTN,IV) = SHPTON
      C
```

SOLUTION ROUTINES - DIMENS SUBROUTINE

```
 ISN
       C  AVERAGE GROUND PRESSURE
 204          P = W / (2.*X4*YGC)
 205          VEHCLE(JGPRES,IV) = P
       C
       C  WT PER LINEAL FOOT
 206          WFT = WTON / (YGC/12.)
 207          VEHCLE(JTONFT,IV) = WFT
       C
       C  MAX DITCH THAT CAN CROSS
 208          DITCH = (YO - (D1/2.) - (D2/2.) - 2.*T) / 24.
 209          VEHCLE(JDWID,IV) = DITCH
       C
       C  MAX SPEED ON LEVEL ROAD
 210          SPD1 = FVAL1(SPEED(1,1),RSHPTN,NSHPTN,SHPTON)
 211          VEHCLE(JMXSPD,IV) = SPD1
       C
       C  MAX SPEED ON 30 DEG GRADE (NO SLIPPAGE)
 212          SPD2 = FVAL1(SPEED(1,2),RSHPTN,NSHPTN,SHPTON)
 213          VEHCLE(J30SPD,IV) = SPD2
       C
       C  MAX SLOPE THAT CAN CLIMB
 214          SLOPMX = FVAL2(SLOPE,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,WTON)
 215          VEHCLE(JSLOPE,IV) = SLOPMX
       C
       C  ACCELERATION SEC FROM 0-20MPH ON LEVEL ROAD
 216          ACC = FVAL2(ACCEL,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,WTON)
 217          VEHCLE(JACCEL,IV) = ACC
       C
       C  RANGE.  FIRST DETERMINE THE TOTAL VOLUME OF FUEL TANKS & THE TOTAL
       C  VOLUME OF PROTECTED FUEL TANKS
 218          IREC = IVECLE(JFUELC,IV)
 219          TFUEL = 0.
 220          FUELP = 0.
 221   45     IF (IREC .EQ. NULL) GO TO 50
 222             GAL = DVAL2(FUELC(JFCAP,IREC),JFUELC,JFCAP,'GAL ')
 223             LOC = IDVAL2(IFUELC(JLOC,IREC),JFUELC,JLOC,'-   ')
```

SOLUTION ROUTINES - DIMENS SUBROUTINE

```
 ISN
224             IF (LOC .LE. 2) FUELP = FUELP + GAL
225             TFUEL = TFUEL + GAL
226             IREC = IFUELC(JNEXT,IREC)
227             GO TO 45
228    50     IENGTP = IDVAL2(IENGIN(JTYPE,LENGIN),JENGIN,JTYPE,'-    ')
229           XNMPG = FVAL2(XMPG(1,1,IENGTP),RSHPTN,NSHPTN,SHPTON,
             +     RVEHWT,NVEHWT,WTON)
230           RANGE = XNMPG * TFUEL
231           RANGEP = XNMPG * FUELP
232           VEHCLE(JRANGE,IV) = RANGE
233           VEHCLE(JRANGA,IV) = RANGEP
       C
       C  ROUGH TERRAIN VELOCITY
234           NUMWHL = IDVAL2(IROADW(JNRDWH,LROADW),JROADW,JNRDWH,'-    ')
235           LSPDMP = IVECLE(JSUSP,IV)
236           WHTRV = DVAL2(SPRING(JWHTRV,LSPDMP),JSUSP,JWHTRV,'IN  ')
237           IF (NUMWHL .LT. MINWHL .OR. NUMWHL .GT. MAXWHL) CALL ERR(41,
             +     'DIMENS',NUMWHL,0,0)
238           NW = NUMWHL - MINWHL + 1
239           VRIDE1 = FVAL3(VRIDE(1,1,1,NW,1),RWHTRV,NWHTRV,WHTRV,
             +     RVEHWT,NVEHWT,WTON,RTKLEN,NTKLEN,YGC)
240           VRIDE2 = FVAL3(VRIDE(1,1,1,NW,2),RWHTRV,NWHTRV,WHTRV,
             +     RVEHWT,NVEHWT,WTON,RTKLEN,NTKLEN,YGC)
       C
       C  PROBABILITY OF VEHICLE BEING PENETRATED BY A HIT
       C
241           CALL INITVL(PPENET(1,1,1,IV),18,FNULL)
242           DO 60 IORD=1,NORDT2
243           DO 60 IASPEC=1,NASPEC
244           DO 60 IRANGE=1,NRANGE
245             PPHULL = FVAL1(PHULLP(1,IRANGE,IASPEC,IORD,IARMHL),RARMTH,
             +     NARMTH,THFU)
246             PPTURR = FVAL1(PTURRP(1,IRANGE,IASPEC,IORD,IARMTU),RARMTH,
             +     NARMTH,TTF)
247             PPENET(IRANGE,IASPEC,IORD,IV) = PHHULL*PPHULL + (1.-PHHULL)*
             +     PPTURR
```

```
                      SOLUTION ROUTINES - DIMENS SUBROUTINE
 ISN
248   60        CONTINUE
249           LVECLE(JPENTR,IV) = IV
250           RETURN
251           END
```

SOLUTION ROUTINES - FPOWER SUBROUTINE

```
ISN
 1          SUBROUTINE FPOWER(IV,IRETN)
      C
      C        ROUTINE TO CALCULATE THE ACCURACY AND LETHALITY OF THE ARMAMENT
      C     SYSTEM SELECTED.  THIS INCLUDES THE PROBABILITY OF HIT AS A FUNCTION
      C     OF RANGE, STATIONARY VS. MOVING FIRER & TARGET, AND ORDNANCE TYPE
      C     FOR THE SAME STANDARD TARGET.  IF THERE ARE ANY CONSTRAINTS ON THESE
      C     VARIABLES SPECIFIED BY THE USER, THE PROGRAM TESTS FOR THEM AND IF
      C     THEY ARE NOT SATISFIED RETURNS AN INDICATOR OF WHICH COMPONENT TYPE
      C     TO BACK UP TO FOR SELECTING AN ALTERNATIVE CHOICE.
      C
      C     INPUT PARAMETERS:
      C        IV       VEHICLE
      C
      C     OUTPUT PARAMETERS:
      C        IRETN    RETURN CODE: 0 IF ALL CONSTRAINTS SATISFIED; OTHERWISE
      C                 SET TO THE INDEX OF THE COMPONENT TYPE TO BACK UP TO.
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
           +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE  (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
ISN
            +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
            +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
            +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
            +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
            +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
  5             EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
  6             COMMON /JPARS/
  7             COMMON /JPARS1/
  8             COMMON /NPARS/
  9             COMMON /AUX/
 10             REAL*8 BLS,DASHES
 11             EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 12             COMMON /ARRAYS/
 13             REAL*8 MANUF,MODEL,NATION
 14             DIMENSION ARRAY(1)
 15             EQUIVALENCE (ARRAY(1),DETRNG(1))
 16             COMMON /JPARS2/
 17             COMMON /DATA/
 18             COMMON /DATA2/
          C
 19             DIMENSION SS(2,3,3)
 20             LOGICAL NOMISL
 21             DATA KSTGT/1/,KMTGT/2/
          C
 22             IRETN = 0
          C
          C  COMPUTE DISPERSION IN THE STATIONARY FIRER - STATIONARY TARGET CASE
          C  BY OBTAINING THE ROOT SUM OF SQUARES OF THE DISPERSIONS ASSOCIATED
          C  WITH EACH OF THE INDIVIDUAL COMPONENTS INVOLVED.
          C
          C  FIRST SET POINTERS TO COMPONENTS INVOLVED.
 23             LMAING = IVECLE(JMAING,IV)
 24             LMISL = IVECLE(JMISL,IV)
 25             LRNGSY = IVECLE(JRNGSY,IV)
 26             LSTBSY = IVECLE(JSTBSY,IV)
 27             LCTLSY = IVECLE(JCTLSY,IV)
```

```
ISN
28            LAMMO = IVECLE(JAMMO,IV)
        C
        C  GET POINTERS TO DISPERSION ARRAYS ASSOCIATED WITH EACH COMPONENT
29            IARRMG = MAINGN(JSIGMA,LMAING)
30            IF (LMISL .NE. NULL) IARRML = MISGUN(JSIGMA,LMISL) + 8
31            IARRRG = IRANGR(JSIGM2,LRNGSY)
32            IARRCL = IGNCTL(JSIGM2,LCTLSY) + 8
        C
        C  DETERMINE PERFORMANCE CLASS OF STABILIZATION SYSTEM
33            ISTBPF = 1
34            IF (LSTBSY .NE. NULL) ISTBPF = ISTABL(JPERF,LSTBSY) + 1
35            IF (ISTBPF .LT. 1 .OR. ISTBPF .GT. 3) CALL ERR(45,'FPOWER',
             +      ISTBPF,1,3)
        C
        C  DETERMINE WHETHER VEHCICLE CAN FIRE MISSILES
36            NOMISL = .TRUE.
        C  IS THERE A MISSILE LAUNCHER ON THE VEHICLE ?
37            IF (LMISL .EQ. NULL) GO TO 3
38               NOMISL = .FALSE.
39               GO TO 10
        C  IS THERE A GUIDED MISSILE ORDNANCE THAT CAN BE FIRED FROM THE
        C  MAIN GUN ?
40      3     CAL = FMAING(JCALIB,LMAING)
41            IR = LAMMO
42      5     IF (IR .EQ. NULL) GO TO 10
43               IF(ABS(AMMO(JCALIB,IR)-CAL) .GT. .001) GO TO 8
44               IF (IAMMO(JGUIDE,IR) .LE. 1) GO TO 8
45                  NOMISL = .FALSE.
46                  GO TO 10
47      8        IR = IAMMO(JNEXT,IR)
48               GO TO 5
        C
        C
        C  ITERATE OVER RELEVANT ORDNANCE TYPES AND RANGES, COMPUTING ROOT
        C  SUM OF SQUARES OF COMPONENTS' DISPERSIONS.
49      10    DO 20 IORDTP=1,NORDTP
```

```
ISN
50              IF (IORDTP .EQ. NORDTP .AND. NOMISL) GO TO 20
51              IARR = IARRMG
52              IF (IORDTP .EQ. NORDTP .AND. LMISL .NE. NULL) IARR = IARRML
53              DO 15 IRANGE=1,NRANGE
54              DO 15 IXY=1,NXY
55                 SS(IXY,IRANGE,IORDTP) = SIGMA(IXY,IRANGE,IORDTP,IARR)**2
           +             + SIGMA2(IXY,IRANGE,IORDTP,IARRRG)**2
           +             + SIGMA2(IXY,IRANGE,IORDTP,IARRCL)**2
56    15           CONTINUE
57    20        CONTINUE
      C
      C  CALCULATE THE PROBABILITY OF HIT AGAINST A STANDARD TARGET FOR
      C  VARIOUS RANGE, ORDNANCE, AND MOVING / STATIONARY CONDITIONS.
      C
58           CALL INITVL(PHITS(1,1,1,IV),18,FNULL)
59           CALL INITVL(PHITM(1,1,1,IV),18,FNULL)
60           CALL INITVL(PK(1,1,1,IV),18,FNULL)
61           DO 40 IORDTP=1,NORDTP
62              IF (IORDTP .EQ. NORDTP .AND. NOMISL) GO TO 40
63              DO 35 IRANGE=1,NRANGE
64                 XSS2 = SS(1,IRANGE,IORDTP)
65                 YSS2 = SS(2,IRANGE,IORDTP)
66                 XSS = SQRT(XSS2)
67                 YSS = SQRT(YSS2)
68                 XMF2 = SIGMVF(1,IRANGE,IORDTP,ISTBPF)
69                 YMF2 = SIGMVF(2,IRANGE,IORDTP,ISTBPF)
70                 XMS = SQRT(XSS2 + XMF2)
71                 YMS = SQRT(YSS2 + YMF2)
72                 XMT2 = SIGMVT(1,IRANGE,IORDTP,1)
73                 YMT2 = SIGMVT(2,IRANGE,IORDTP,1)
74                 XSM = SQRT(XSS2 + XMT2)
75                 YSM = SQRT(YSS2 + YMT2)
76                 XMM = SQRT(XSS2 + XMF2 + XMT2)
77                 YMM = SQRT(YSS2 + YMF2 + YMT2)
78                 PHITS(IRANGE,KSTGT,IORDTP,IV) = PHIT(XSS,YSS)
79                 PHITS(IRANGE,KMTGT,IORDTP,IV) = PHIT(XSM,YSM)
```

```
ISN
80                    PHITM(IRANGE,KSTGT,IORDTP,IV) = PHIT(XMS,YMS)
81                    PHITM(IRANGE,KMTGT,IORDTP,IV) = PHIT(XMM,YMM)
      C
      C  CALCULATE THE PROBABILITY OF A KILL GIVEN A HIT AGAINST THE TARGET
82                    DO 30 IASPEC = 1,NASPEC
83                       PK(IRANGE,IASPEC,IORDTP,IV) = FVAL1(PKDAT(1,IRANGE,
           +                IASPEC,IORDTP),RDISPR,NDISPR,(XSS+YSS)/2.)
84    30                 CONTINUE
85    35              CONTINUE
86    40           CONTINUE
87              IVECLE(JPHITS,IV) = IV
88              IVECLE(JPHITM,IV) = IV
89              IVECLE(JPK,IV) = IV
90              RETURN
91              END
```

```
                 SOLUTICN ROUTINES - GENVEH SUBROUTINE
ISN
 1          SUBROUTINE GENVEH
      C
      C     ROUTINE TO GENERATE THE "BEST" CONCEPT VEHICLE FROM THE COMPONENTS
      C  AVAILABLE AND MATCHING THE CONSTRAINTS SPECIFIED.
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +     ICRFW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFTREX(1)),(ENVIRC(1),
          +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6          COMMON /VPROTO/
 7          REAL*8 FLDNAM,RECNAM,DFAULT
 8          COMMON /JPARS/
 9          COMMON /JPARS1/
10          COMMON /NPARS/
11          COMMON /AUX/
```

```
                   SCLUTICN ROUTINES - GENVEH SUBROUTINE
ISN
12            REAL*8 BL8,DASHES
13            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
14            DIMENSION JJ(35),JCT(35),NUM(35),MATCH(35),NUMSPC(35)
15            LOGTCAL OKCMP,COMPAT
      C
      C  ORDER THE COMPONENTS OF EACH TYPE IN DECREASING ORDER BY SCORE
16            CALI CSCORE
      C
      C  DETERMINE NUMBER OF USER SPECIFICATIONS FOR EACH COMPONENT
      C
17            CALL IZERO(NUMSPC,NCMPTP)
18            DO 2 ICMP=1,NCMPTP
19               IF (IVECLE(ICMP,ISPECV) .EQ. NULL) GO TO 2
20               IOFF = IFILPT(ICMP+1)
21               NUMSPC(ICMP) = NLIST(VFILE(IOFF),NFLDS(ICMP+1),JNEXT,
             +     TVECLE(ICMP,ISPECV))
22    2          CONTINUE
      C
      C  --  VEHICLE SPECIFICATION FILE CONTAINS INFORMATION
      C  --  ABOUT THE VEHICLE MODEL WHICH INTERFERES WITH
      C  --  THE MATCHING OF SPECIFICATION TO ALTERNATIVES COMPONENTS.
      C  --  THEREFORE,  THE JMODEL FIELD MUST BE NULL BEFORE
      C  --  THE BEST VEICLE CAN BE GENERATED.
      C
23            DO 4 ICMP=1,NCMPTP
24            IR = IVECLE(ICMP,ISPECV)
25          3 IF(IR .EQ. NULL) GO TO 4
26            IOFF = IFILPT(ICMP+1)
27            NF = NFLDS(ICMP+1)
28            CALL STORE(NULL,IVFILE(IOFF),NF,JMODEL,IR)
29            IR = IVAL(TVFILE(IOFF),NF,JNEXT,IR)
30            GO TO 3
31          4 CONTINUE
      C
      C
```

```
                   SCLUTION ROUTINES - GENVEH SUBROUTINE
ISN
        C   ITERATE OVER COMPONENT TYPES IN ORDER THAT WISH TO PROCESS THEM
        C
32             I = 1
33      5      ICMP = ICMPRK(I)
34             IRECTP = ICMP + 1
35             NF = NFLDS(IRECTP)
36             IOFF = IFILPT(IRECTP)
        C
        C   ITERATE OVER COMPONENTS OF THIS TYPE (THEY ARE IN APPROPRIATE ORDER
        C   ALREADY)
        C
37             JI = ICMPPT(ICMP)
38             JJ(I) = JI
39             JCT(I) = 0
40             NUM(I) = 0
41             MATCH(I) = 0
        C
        C   DOES COMPONENT JI SATISFY USFR SPECS ?
        C
42             IF (NUMSPC(TCMP) .EQ. 0 .AND. MINNUM(ICMP) .EQ. 0) GO TO 10
43      8       IF (NUMSPC(ICMP) .GT. 0 .AND.
              +  .NOT.OKCMP(ICMP,JI,VFILE(IOFF),NF,MATCH(T),NUM(I),NUMSPC(ICMP)))
              +      GO TO 12
        C
        C   TS COMPONENT JI COMPATIBLE WITH PREVIOUSLY SELECTED COMPONENTS ?
        C
44             IF (.NOT.COMPAT(ICMP,JI)) GO TO 15
        C
        C   REMEMBER COMPONENT JI AS PART OF SPECIFICATION VEHICLE
        C
45             CALL QUEUE(JI,IVECLE(ICMP,ICANDV),IRECTP)
        C
        C   HAS MINIMUM NUMBER OF COMPONENTS OF THIS TYPE BEEN SELECTED ?
        C   IF NOT, GO GET ANOTHER.
        C
46             IF (NUM(I) .LT. MINNUM(ICMP)) GO TO 15
```

```
ISN
      C
      C  CALCULATE ANY PERFORMANCE PARS APPROPRIATE AFTER SELECTION OF THIS
      C  COMPONENT TYPE
      C
47          IRETN = IPFORM(ICMP)
48          IF (IRETN .EQ. 0) GO TO 10
      C
      C  IF BACKTRACKING IS INDICATED, DO SO
      C
49             IF (IRETN .GE. 1) CALL ERR(44,'GENVEH',I,IRETN,ICMP)
50             DO 11 K=IRETN,1
51                CALL UNHOOK(ICMPRK(K))
52     11         CONTINUE
53             I = IRETN
54             GO TO 22
      C
      C  PROCESS NEXT COMPONENT TYPE IF THERE IS ONE
      C
55     10   I = I + 1
56          IF (I .GT. NCMPTP) RETURN
57          GO TO 5
      C
      C  COMPONENT DIDN'T SATISFY USER SPECS.  KEEP NOTE OF NUMBER OF SUCH
      C  COMPONENTS.
      C
58     12   JCT(I) = JCT(I) + 1
      C
      C  GET NEXT COMPONENT OF CURRENT TYPE, IF THERE IS ONE
59     15   JI = IVAL(VFILE(IOFF),NF,JSCORE,JI)
60          JJ(I) = JI
61          IF (JI .NE. NULL) GO TO 8
      C
      C  CAN FAILURE TO FIND A SUITABLE COMPONENT POSSIBLY BE AVOIDED BY
      C  BACKTRACKING ?
      C
62          IF (JCT(I)+MINNUM(ICMP) .LE. NALTCS(ICMP)) GO TO 20
```

SOLUTION ROUTINES - GENVEH SUBROUTINE

```
ISN
63               CALL ERR(42,'GENVEH',ICMP,JI,JCT(I))
64               PAUSE
      C
      C  BACKTRACK TO PREVIOUS COMPONENT TYPE
      C
65    20      CALL UNHOOK(ICMP)
66            I = I - 1
67    22      IF (I .GE. 1) GO TO 25
68               CALL ERR(43,'GENVEH',ICMP,0,0)
69               PAUSE
      C
      C  TRY NEXT CHOICE OF PREVIOUS COMPONENT TYPE
      C
70    25      ICMP = ICMPRK(I)
71            IRECTP = ICMP + 1
72            NF = NFLDS(IRECTP)
73            IOFF = IFILPT(IRECTP)
74            JI = TVAL(VFILE(IOFF),NF,JSCORE,JJ(I))
75            IF (JI .EQ. NULL) GO TO 20
76            NUM(I) = 0
77            CALL UNHOOK(ICMP)
78            MATCH(I) = 0
79            JJ(I) = JI
80            GO TO 8
      C
81    END
82            FUNCTION IPFORM(ICMP)
      C
      C       ROUTINE TO SUPERVISE THE CALCULATION OF PERFORMANCE VARIABLES.
      C  THE ROUTINE IS CALLED AFTER EACH COMPONENT OF A GIVEN TYPE IS
      C  SELECTED FOR THE CANDIDATE VEHICLE.  IF THIS IS AN APPROPRIATE TIME
      C  TO CALCULATE A GIVEN PERFORMANCE PARAMETER (I.E., ALL RELEVANT
      C  INDEPENDENT VARIABLES ARE KNOWN AT THIS POINT OR CAN BE ESTIMATED
      C  SUFFICIENTLY WELL TO TEST A CONSTRAINT ON A PERFORMANCE PARAMETER),
      C  THE ROUTINE DOES SO.  WHENEVER THE ROUTINE CALCULATES A PERFORMANCE
      C  PARAMETER IT CHECKS WHETHER IT SATISFIES ANY CONSTRAINTS THAT MAY BE
```

```
ISN
      C  IMPOSED ON IT, EITHER BY THE USER OR FROM THE PROGRAM'S KNOWLEDGE OF
      C  VEHCILE DESIGN.  IF A PERFORMANCE PARAMETER DOES NOT MEET AN IMPOSED
      C  CONSTRAINT, THE CODE INDICATES HOW FAR TO BACKUP TO TRY TO OBTAIN A
      C  BETTER MATCH (I.E., WHAT COMPONENT TYPE TO RETURN TO TO TRY TO SELECT
      C  A DIFFERENT CHOICE).  THE FUNCTION RETURNS THE INDEX OF SUCH A
      C  COMPONENT IF ANY TESTED CONSTRAINTS FAIL, OR IT RETURNS ZERO
      C  OTHERWISE.
      C
      C  INPUT PARAMETERS:
      C     ICMP    TYPE OF COMPONENT JUST SELECTED FRO CANDIDATE VEHICLE
      C
 83             COMMON /VEH/
 84             DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
              +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
              +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
              +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
              +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
              +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
              +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
              +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
              +     IRELN(3,50),VFILE(1),IVFILE(1)
 85             EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
              +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
              +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
              +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
              +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
              +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
              +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
              +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
              +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
              +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
              +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
              +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
              +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 86             EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 87             COMMON /JPARS/
```

```
ISN
 88           COMMON /JPARS1/
 89           COMMON /AUX/
 90           REAL*8 BL8,DASHES
 91           EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 92           COMMON /SPECS/
 93           REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
      C
 94           LOGICAL OKVAL
      C
 95           IPFORM = 0
 96           GO TO (10,20,10,10,10,  10,10,10,10,10,  60,10,80,10,100,
             +     10,10,10,10,10,  10,10,10,10,10,  10,10,10,10,10,  160),ICMP
      C
 97   10      RETURN
      C
      C  HAVE JUST SELECTED TURRET.  CAN NOW COMPUTE THE GEOMETRY, WEIGHT,
      C  PROTECTION, AND MANY OF THE MOBILITY PARAMETERS IN FINAL FORM.
 98      20 CALL DIMENS(ICANDV,IPFORM)
 99           RETURN
      C
      C  HAVE JUST SELECTED ARMAMENT COMPONENTS.  COMPUTE ACCURACY &
      C  LETHALITY PARAMETERS AND CHECK THAT THEY SATISFY ANY USER-PROVIDED
      C  CONSTRAINTS.
100   60      CALL FPOWER(ICANDV,IPFORM)
101           RETURN
      C
      C  HAVE JUST SELECTED THE ENGINE.  CAN MAKE QUICK TEST THAT HAVE
      C  SUFFICIENT GROSS HP / TON.
102   80      IF (VEHCLE(JGHPTN,ISPECV) .EQ. FNULL) RETURN
103           LENGIN = IVECLE(JENGIN,ISPECV)
104           HP = ENGINE(JHP,LENGIN)
105           IF (OKVAL(HP/VEHWT,1,JGHPTN,NULL)) RETURN
106           IPFORM = ICMP
107           RETURN
      C
      C  HAVE JUST SELECTED THE FINAL DRIVE.  CAN MAKE INITIAL ESTIMATE OF
```

SOLUTION ROUTINES - GENVEH SUBROUTINE

```
 ISN
        C  POWER LIMITED MOBILITY PARAMETERS & BACKTRACK IF THESE DON'T MEET
        C  CONSTRAINTS.
108     100    CALL MOBILE(ICANDV,IPFORM)
109            RETURN
        C
        C  HAVE SELECTED ALL COMPONENTS.  CHECK RAM/D & COST CONSTRAINTS.
        C  (THIS PORTION NOT IMPLEMENTED).
110     160    RETURN
111            END
```

SOLUTION ROUTINES - MOBILE SUBROUTINE

```
ISN
 1          SUBROUTINE MOBILE(IV,IRETN)
      C
      C        ROUTINE TO CALCULATE SELECTED POWER-CONSTRAINED MOBILITY
      C     PARAMETRS TO SEE WHETHER THE CHOICE OF POWER TRAIN COMPONENTS IS
      C     SATISFACTORY IF THE USER HAS SPECIFIED ANY CONSTRAINTS ON THESE
      C     PARAMETERS.
      C
      C     INPUT PARAMETERS:
      C        IV       VEHICLE OF INTEREST
      C
      C     OUTPUT PARAMETERS:
      C        IRETN    COMPONENT RANK INDEX TO BACK UP TO IF THE USER'S
      C                 CONSTRAINTS ARE NOT SATISFIED; 0 OTHERWISE.
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +      IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIFT(1)),
          +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
                 SOLUTION ROUTINES - MOBILE SUBROUTINE
ISN
          +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
  5        EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
  6        COMMON /VPROTO/
  7        REAL*8 FLDNAM,RECNAM,DFAULT
  8        COMMON /JPARS/
  9        COMMON /JPARS1/
 10        COMMON /NPARS/
 11        COMMON /AUX/
 12        REAL*8 BL8,DASHES
 13        EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 14        COMMON /SPECS/
 15        REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
 16        COMMON /DATA/
 17        COMMON /DATA2/
      C
 18        LOGICAL OKVAL
      C
 19        IRETN = 0
      C
 20        LENGIN = IVECLE(JENGIN,IV)
 21        LTRANS = IVECLE(JTRANS,IV)
 22        LFINDR = IVECLE(JFINDR,IV)
      C
      C  CHECK GROSS HP / TON
 23        HP = DVAL2(ENGINE(JHP,LENGIN),JENGIN,JHP,'HP  ')
 24        HPTON = HP / VEHWT
 25        IF (.NOT.OKVAL(HPTON,1,JGHPTN,NULL)) GO TO 999
      C
      C  CHECK SPROCKET HP / TON
 26        EFFTR = DVAL2(TRANSM(JEFFIC,LTRANS),JTRANS,JEFFIC,'-')
 27        EFFFD = DVAL2(FINLDR(JEFFIC,LFINDR),JFINDR,JEFFIC,'-')
 28        SHPTON = HP * EFFTR * EFFFD / VEHWT
 29        IF (.NOT.OKVAL(SHPTON,1,JSHPTN,NULL)) GO TO 999
      C
      C  CHECK SPEED ON LEVEL ROAD IF CONSTRAINED
```

```
           IF (IVECLE(JMXSPD,ISPECV) .EQ. NULL) GO TO 10
           SPD1 = FVAL1(SPEED(1,1),RSHPTN,NSHPTN,SHPTON)
           IF (.NOT.OKVAL(SPD1,1,JMXSPD,NULL)) GO TO 999
C
C  CHECK SPEED ON 30 DEGREE GRADE, IF CONSTRAINED
   10      IF (IVECLE(J30SPD,ISPECV) .EQ. NULL) GO TO 20
           SPD2 = FVAL1(SPEED(1,2),RSHPTN,NSHPTN,SHPTON)
           IF (.NOT.OKVAL(SPD2,1,J30SPD,NULL)) GO TO 999
C
C  CHECK MAX SLOPE TJAT TANK CAN CLIMB, IF THIS IS CONSTRAINED
   20      IF (IVECLE(JSLOPE,ISPECV) .EQ. NULL) GO TO 30
           SLOPMX = FVAL2(SLOPE,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,VEHWT)
           IF (.NOT.OKVAL(SLOPMX,1,JSLOPE,NULL)) GO TO 999
C
C  CHECK MAX ACCELERATION THAT VEHICLE CAN ACHIEVE, IF CONSTRAINED
   30      IF (IVECLE(JACCEL,ISPECV) .EQ. NULL) RETURN
           ACC = FVAL2(ACCEL,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,VEHWT)
           IF (.NOT.OKVAL(ACC,1,JACCEL,NULL)) GO TO 999
           RETURN
C
C  HAVE FOUND AT LEAST ONE POWER-LIMITED MOBILITY PARAMETER CONSTRAINT
C  THAT WAS NOT SATISFIED.  BACKUP ONE.  TRY NEW FINAL DRIVE.
  999      DO 1000 IRETN=1,NCMPTP
              IF (ICMPRK(IRETN) .EQ. JFINDR) RETURN
 1000         CONTINUE
           PAUSE
           END
```

SOLUTION ROUTINES - FUNCTION OKCMP

```
ISN
 1           LOGICAL FUNCTION OKCMP(ICMP,JI,IFILE,NF,MATCH,NUM,NUMSPC)
       C
       C        ROUTINE TO TEST WHETHER A SELECTED COMPONENT IS COMPATIBLE WITH
       C     THE USER SPECIFICATIONS.  THE SELECTED COMPONENT IS DEEMED COMPATIBLE
       C     WITH THE USER SPECIFICATIONS IF EITHER OF THE FOLLOWING CONDITIONS
       C     ARE MET:  (1) THE USER HAS INDICATED A SPECIFIC COMPONENT TO BE
       C     SELECTED AND THE COMPONENT IN QUESTION IS THAT COMPONENT, AND
       C     (2) THERE IS AT LEAST ONE COMPONENT OF THE CURRENT TYPE THAT MAY BE
       C     SELECTED WITHOUT EXCEEDING THE MAX NUMBER OF COMPONENT CONSTRAINTS
       C     AND IF THERE ARE ANY SETS OF ENGINEERING CONSTRAINTS SPECIFIED
       C     WHICH HAVEN'T YET BEEN PAIRED WITH A SPECIFIC COMPONENT SELECTION,
       C     ONE SUCH SET CAN BE PAIRED WITH THE CURRENT SELECTION SUCH THAT NO
       C     CONSTRAINTS ARE VIOLATED.  IN THESE CASES THE FUNCTION RETURNS A
       C     .TRUE. VALUE; OTHERWISE IT RETURNS .FALSE.
       C
       C     INPUT PARAMETERS:
       C        ICMP     COMPONENT TYPE
       C        JI       TRIAL COMPONENT TO BE TESTED FOR COMPATIBILITY
       C        IFILE    FILE IN WHICH COMPONENTS ARE LOCATED
       C        NF       NUMBER OF FIELDS IN RECORDS OF THIS FILE
       C        MATCH    NUMBER OF MATCHES OBTAINED BETWEEN COMPONENTS OF THIS
       C                 TYPE SPECIFIED BY THE USER AND COMPONENTS SELECTED SO FAR
       C        NUM      NUMBER OF COMPONENTS OF THIS TYPE ALREADY SELECTED
       C        NUMSPC   NUMBER OF COMPONENTS OF THIS TYPE REFERRED TO BY THE
       C                 USER SPECIFICATIONS
       C
       C     OUTPUT PARAMETERS:
       C        MATCH    UPDATED BY 1 IF THE CANDIDATE COMPONENT CAN BE PAIRED
       C                 WITH AN AS YET UNMATCHED USER SPECIFICATION
       C        NUM      UPDATED BY 1 IF THE CANDIDATE COMPONENT IS ACCEPTABLE.
       C
 2           COMMON /VEH/
 3           DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
```

SOLUTION ROUTINES - FUNCTION OKCMP

```
ISN
               +       IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
               +       IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
               +       IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
               +       IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
               +       IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
               +       IRELN(3,50),VFILE(1),IVFILE(1)
   4            EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
               +       ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
               +       (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
               +       IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
               +       (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
               +       IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
               +       (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
               +       ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
               +       (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
               +       ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
               +       (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
               +       IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
               +       (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
   5            EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
   6            COMMON /VPROTO/
   7            REAL*8 FLDNAM,RECNAM,DFAULT
   8            COMMON /JPARS/
   9            COMMON /JPARS1/
  10            COMMON /NPAES/
  11            COMMON /AUX/
  12            REAL*8 BL8,DASHES
  13            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
  14            EQUIVALENCE (JMATCH,JMODEL)
      C   USE AN UNUSED FIELD TO POINT FROM A USER SPEC RECORD TO THE
      C   ACTUAL COMPONENT PAIRED WITH IT.
      C
  15            DIMENSION IFILE(NF,1)
  16            LOGICAL OKVAL,IOKVAL
  17            DATA NCMPMX/50/
```

SOLUTION ROUTINES - FUNCTION OKCMP

```
ISN
       C
18             OKCMP = .TRUE.
19             IRECTP = ICMP + 1
       C
       C   ARE THERE ANY USER SPECIFICATIONS ?
20             IF (IVECLE(ICMP,ISPECV) .EQ. NULL)   NUM=NUM+1
21             IF (IVECLE(ICMP,ISPECV) .EQ. NULL) RETURN
       C
       C   IS THE CURRENT COMPONENT SPECIFICALLY REQUESTED BY THE USER ?
22             IR = INLIST(IFILE(JID,JI),IFILE,NF,JID,JNEXT,IVECLE(ICMP,ISPECV))
23             IF (IR .EQ. NULL) GO TO 10
24                MATCH = MATCH + 1
25                NUM = NUM + 1
26                IFILE(JMATCH,IR) = JI
27                RETURN
       C
       C   ARE THERE ALREADY TOO MANY COMPONENTS ?
28     10      IF (NUM + (NUMSPC - MATCH) .LE. MAXNUM(ICMP)) GO TO 12
29                CKCMP = .FALSE.
30                RETURN
       C
       C   TTERATE OVER ANY UNMATCHED SETS OF ENGINEERING PARAMETER CONSTRAINTS
31     12      IR = IVECLE(ICMP,ISPECV)
       C   SKIP OVER ANY SPECIFICATIONS THAT REFER TO CONSTANT COMPONENTS
32     14      IF (IFILE(JID,IR) .LT. NCMPMX) GO TO 30
       C
       C   SKIP OVER ANY SPECIFICATIONS THAT HAVE ALREADY BEEN MATCHED TO
       C   ANOTHER COMPONENT
33             IF (IFILE(JMATCH,IR) .NE. NULL) GO TO 30
       C   DOES THE TRIAL COMPONENT SATISFY THIS SET ?
       C
34             IOFFR = IFILPT(ICMP+1)
35             DO 20 IFLD=2,NF
36                IF (IFLD .EQ. JID) GO TO 20
37                IVF = IVAL(IVFILE(IOFFR),NF,IFLD,IR)
38                IF (IVF .EQ. NULL) GO TO 20
```

SOLUTION ROUTINES - FUNCTION OKCMP

```
ISN
      C  HAVE FOUND A CONSTRAINT.  TEST IT.  FIRST DETERMINE TYPE OF FIELD.
39                KFLD = IFLDPT(IRECTP) + IFLD
40                ITYP = IFLDTP(KFLD)
      C  CONSTRAINTS ON ARRAYS OR CHARACTER FIELDS ARE NOT IMPLEMENTED
41                IF (ITYP .EQ. ITYPES(3) .OR. ITYP .EQ. ITYPES(4)) GO TO 20
      C  IS TYPE REAL ?
42                IF (ITYP .NE. ITYPES(1)) GO TO 15
      C  IF SO GET VALUE
43                  V = DVAL2(IFILE(IFLD,JI),ICMP,IFLD,UNITS(KFLD))
44                  IF (.NOT.OKVAL(V,IRECTP,IFLD,IR)) GO TO 30
45                  GO TO 20
      C  TYPE IS INTEGER
46    15          IV = IFILE(IFLD,JI)
47                IF (IV .EQ. NULL) IV = DFAULT(KFLD)
48                IF (.NOT.IOKVAL(IV,IRECTP,IFLD,IR)) GO TO 30
49    20          CONTINUE
      C
      C  COMPONENT DOES INDEED SATISFY THIS SET OF ENGINEERING CONSTRAINTS
50            NUM = NUM + 1
51            MATCH = MATCH + 1
52            IFILE(JMATCH,IR) = JI
53            RETURN
      C
      C  GET NEXT USER SPECIFICATION
54    30      IR = IFILE(JNEXT,IR)
55            IF (IR .NE. NULL) GO TO 14
      C
      C  COMPONENT INCOMPATIBLE WITH ENGINEERING CONSTRAINTS
56    40      OKCMP = .FALSE.
57            RETURN
58            END
```

SOLUTION ROUTINES - PHIT FUNCTION

```
ISN
 1          FUNCTION PHIT(XSIG,YSIG)
      C
      C        ROUTINE TO CALCULATE THE PROBABILITY OF A HIT AGAINST A TARGET OF
      C     A STANDARD SIZE AND SHAPE HAVING A SPECIFIED AIM POINT, GIVEN A
      C     STANDARD DEVIATION OF THE FIRE IN THE X AND Y DIRECTIONS.  THE
      C     DISPERSION OF FIRE IS ASSUMED TO FOLLOW A BIVARIATE NORMAL
      C     DISTRIBUTION.  THIS VERSION OF THE ROUTINE ASSUMES A SQUARE TARGET
      C     2.34 M ON A SIDE WITH THE AIM POINT IN THE MIDDLE.  THE STANDARD
      C     DEVIATIONS OF THE FIRING DISTRIBUTION IS ALSO ASSUMED TO BE
      C     EXPRESSED IN METERS.
      C
 2          DATA RSQRT2/.70711/,XTGT/1.17/,YTGT/1.17/
      C
 3          IF (XSIG .LT. 1.E-6) XSIG = 1.E-6
 4          IF (YSIG .LT. 1.E-6) YSIG = 1.E-6
 5          PHIT = ERF(RSQRT2*XTGT/XSIG) * ERF(RSQRT2*YTGT/YSIG)
 6          RETURN
 7          END
```

C.5: OUTPUT ROUTINES

This section contains listings of routines used to format and write the output tables which display the results of a model run.

OUTPLT ROUTINES - OUTARR SUBROUTINE

```
ISN
 1           SUBROUTINE OUTARR(ARR,IRECTP,IFLD)
      C
      C        ROUTINE TO OUTPUT TFE VALUES ASSOCIATED WITH AN ARRAY WHICH
      C    IS THE VALUE OF A RECORD'S FIELD
      C
      C    INPUT PARAMETERS:
      C        ARR      ARRAY OF INTEREST
      C        IRECTP   RECORD TYPE OF RECORD POINTING TO ARRAY
      C        IREC     RECORD POINTING TO ARRAY
      C        IFLD     FIELC CF RECORD POINTING TO ARRAY
      C
 2           CCMMCN /VPROTO/
 3           RFAL*8 FLDNAM,RECNAM,DFAULT
 4           COMMON /NPARS/
 5           CCMMON /AUX/
 6           REAL*8 BL8,CASHES
 7           EOUIVALENCF (FNULL,NULL),(IBLS,BLS)
 8           CCMMCN /ARRAYS/
 9           REAL*8 MANUF,MODEL,NATICN
10           DIMENSION ARRAY(1)
11           EOUIVALENCF (ARRAY(1),DETPNG(1))
12           COMMON /JPARS2/
      C
13           DIMENSION ARR(1),ILEVS(3),NLEVS(3)
      C
      C    GET PARAMETERS OF INTEREST ASSOCIATED WITH ARRAY
14           ND = 0
15           DO 10 I=1,NADIMS
16              IF (IARDEF(JRECTP,I) .NE. IRECTP .OR. IARDEF(JFLDRC,I) .NE.
           +      IFLD) GC TO 10
17              ND = ND + 1
18              NLEVS(ND) = IARDEF(JNLEVL,I)
19              ILEVS(ND) = IARDEF(JLEVPT,I)
20    10        CONTINUE
21           IFLDRC = IFLCPT(IRECTP) + IFLD
```

```
                    OUTPUT ROUTINES - OUTARR SUBROUTINE
ISN
22            UNIT = UNITS(IFLCRC)
      C
      C  BRANCH ACCORDING TO NUMBER OF DIMENSIONS OF ARRAY
23            IF (ND .LE. 0 .OR. ND .GT. 3) CALL ERR(38,'OUTARR',ND,0,0)
24            GO TO (20,40,60), ND
      C
      C  1-DIMENSIONAL ARRAY.  ITERATE OVER LEVELS OF ARRAY, OUTPUTTING
      C  EACH WITH ASSOCIATED VALUES
25    20      N = NLEVS(1)
26            DO 30 I=1,N
27               K = ILEVS(1) + I - 1
28            IF(ARR(I) .EQ. FNULL) GO TO 31
29               WRITE (7,100) (ARRLAB(J,K),J=1,5),ARR(I),UNIT
30            GO TO 30
31    100        FORMAT (4X,5A4,10X,F10.2,4X,A4)
32    31         WRITE(7,101) (ARRLAB(J,K),J=1,5)
33    101     FORMAT(4X,5A4,16X,'N/A')
34    30         CONTINUE
35            RETURN
      C
      C  2-DIMENSIONAL ARRAY.  ITERATE OVER DIMENSIONS IN REVERSE ORDER.
      C  ITERATE OVER LEVELS OF ARRAY, OUTPUTTING EACH WITH ASSOCAITED VALUE
36    40      L = 0
37            N2 = NLEVS(2)
38            DO 50 JD=1,N2
39               K2 = ILEVS(2) + JD -1
40               WRITE(7,100) (ARRLAB(J,K2),J=1,5)
41               N1 = NLEVS(1)
42               DO 50 ID=1,N1
43                  L = L + 1
44                  K1 = ILEVS(1) + ID -1
45              IF(ARR(L) .EQ. FNULL) GO TO 51
46                  WRITE (7,105) (ARRLAB(J,K1),J=1,5),ARR(L),UNIT
47            GO TO 50
48    105           FORMAT (7X,5A4,7X,F10.2,4X,A4)
49    51       WRITE(7,106) (ARRLAB(J,K1),J=1,5)
```

OUTPUT ROUTINES - OUTARR SUBROUTINE

```
ISN
50    106    FORMAT(7X,5A4,13X,'N/A')
51    50           CONTINUE
52           RETURN
      C
      C  3-DIMENSIONAL ARRAY.  ITERATE OVER DIMENSIONS IN REVERSE ORDER.
      C  ITERATE OVER LEVELS OF ARRAY, FOR EACH OUTPUTTING LABEL & VLUE
53    60     L = 0
54           N3 = NLEVS(3)
55           DO 70 KD=1,N3
56              K3 = ILEVS(3) + KD - 1
57              WRITE (7,100) (ARRLAB(J,K3),J=1,5)
58              N2 = NLEVS(2)
59              DO 70 JD=1,N2
60                 K2 = ILEVS(2) + JD - 1
61                 WRITE (7,105) (ARRLAB(J,K2),J=1,5)
62                 N1 = NLEVS(1)
63                 DO 70 ID=1,N1
64                    L = L + 1
65                 IF(ARR(L) .EQ. FNULL) GO TO 71
66                    K1 = ILEVS(1) + ID - 1
67                    WRITE (7,110) (ARRLAB(J,K1),J=1,5),ARR(L),UNIT
68                 GO TO 70
69    110             FORMAT (10X,5A4,4X,F10.2,4X,A4)
70    71       WRITE(7,111) (ARRLAB(J,K1),J=1,5)
71    111    FORMAT(10X,5A4,10X,'N/A')
72    70              CONTINUE
73           RETURN
74           END
```

```
                      OUTPUT RCUTINES - CUTCOM SUBROUTINE
ISN
 1              SUBROUTINE OUTCOM
       C
       C        ROUTINE TO OUTPUT THE ATTRIBUTES IN COMMON TO ALL COMPONENTS OF
       C  THE VEHICLE AS WELL AS COMBINED MEASURES OFR THE VEHICLE AS A WHOLE.
       C
 2              COMMON /VPROTO/
 3              REAL*8 FLDNAM,RECNAM,DFAULT
 4              COMMON /JPARS/
 5              COMMON /JPARS1/
 6              COMMON /NPARS/
 7              COMMON /OUTPLT/
 8              LOGICAL*1 DF
 9              REAL*8 XNAT,XMANUF,XMODEL
       C
10              DIMENSION ISUB(3),CV(5)
11              DATA ISUB/'A','B','C'/,CV/'    C','ONCE','PT V','EHIC','LE'/
       C
       C  PRODUCE TABLE IA: VEHICLE COMPONENT IDENTIFICATIONS & PRODUCIBILITY
       C
       C  OUTPUT TABLE TITLE
12              WRITE (7,100) ISUB(1)
13     100      FORMAT ('1',23X,'TABLE I',A1,'.   CONCEPT VEHICLE:',
           +        ' ATTRIBUTES IN COMMON TO ALL COMPONENTS.')
       C
       C  OUTPUT TABLE COLUMN HEADERS
14              WRITE (7,105)
15     105      FORMAT ('-',32X,'-------   IDENTIFICATION   -------',10X,
           +        'PRODUCIBILITY'/' SUBSYSTEM / COMPONENT',10X,'NATION','     ',
           +        'MANUF      MODEL     COMP INDX    # PROD   R&D TIME')
       C
       C  ITERATE OVER COMPCNENT LABELS, WRITING LABELS & ASSOCIATED VARIABLES
16              LAST = IPPARS - 1
17              KK = 0
18              DO 20 I=1,LAST
19                 IF (IOUTF(I) .NE. 0) GO TO 10
```

OUTPUT ROUTINES - OUTCOM SUBROUTINE

```
ISN
      C  WRITE SPACER OR SUBSYSTEM HEADER WITHOUT CORRESPONDING VALUES
 20              WRITE (7,110) (OUTNAM(J,I),J=1,7)
 21   110        FORMAT (1X,7A4)
 22              GO TO 20
      C  WRITE LABEL & VALUES
 23   10         KK = KK + 1
 24              K = KOUT(KK)
 25              WRITE (7,115) (OUTNAM(J,I),J=1,7),XNAT(K),XMANUF(K),XMODEL(K),
          +        INDX(K),NPROD(K),DF(1,K),RDTIME(K),DF(2,K)
 26   115        FORMAT (1X,7A4,3X,A8,2(2X,A8),4X,I2,6X,I6,A1,2X,F6.1,A1)
 27   20         CONTINUE
      C
      C  OUTPUT SEPARATOR LINE & VEHICLE SUMMARY
      C     WRITE (7,120) CV,NPROD(NCMPT1)
 28   120   FORMAT (82X,'______'/1X,5A4,60X,F6.1)
      C
      C  NOW DO TABLE IB:  NUMBER, WEIGHT, VOLUME, & RAM/D
      C
      C  OUTPUT TABLE TITLE
 29         WRITE (7,100) ISUB(2)
      C
      C  OUTPUT TABLE CCLUMN HEADERS
 30         WRITE (7,125)
 31   125   FORMAT ('-',47X,'WEIGHT',9X,'INTERNAL VOLUME',6X,'----------    ',
          +      'R A M / D  ---------'/1X,' SUBSYSTEM / COMPONENT',10X,
          +      'NUMBER',7X,'LBS  PERCENT',6X,'CU FT  PERCENT',5X,
          +      'MAINT/OP  MATURITY  CMPLXITY  MMBF')
      C
      C  ITERATE OVER COMPONENT LABELS, WRITING LABELS & ASSOCIATED VARIABLES
 32         KK = 0
 33         DO 40 I=1,LAST
 34           IF (IOUTF(I) .NE. 0) GO TO 30
      C  WRITE SPACER OR SUBSYSTEM HEADER WITHOUT CORRESPONDING VALUES
 35             WRITE (7,110) (OUTNAM(J,I),J=1,7)
 36             GO TO 40
      C  WRITE LABEL & VALUES
```

OUTPUT ROUTINES - OUTCOM SUBROUTINE

```
ISN
37    30      KK = KK + 1
38             K = KOUT(KK)
39            IF( K .EQ. JHULL .OR. K .EQ. JTURET) GO TO 35
40               WRITE (7,130) (OUTNAM(J,I),J=1,7),NUMBER(K),WEIGHT(K),
          +        DF(3,K),PCWT(K),VOLUME(K),DF(4,K),PCVOL(K),DWNUP(K),
          +        DF(5,K),MATUR(K),DF(6,K),ICMPLX(K),DF(7,K),RELIAB(K),DF(8,K)
41            GO TO 40
42    35      WRITE(7,131) (OUTNAM(J,I),J=1,7),NUMBER(K),WEIGHT(K),
          +      DF(3,K),PCWT(K),VOLUME(K),DF(4,K),DWNUP(K),
          +      DF(5,K),MATUR(K),DF(6,K),ICMPLX(K),DF(7,K),RELIAB(K),DF(8,K)
43    130        FORMAT (1X,7A4,5X,I2,4X,F9.2,A1,3X,F5.1,5X,F6.1,A1,3X,F5.1,5X,
          +        F6.1,A1,6X,I1,A1,8X,I1,A1,3X,F7.1,A1)
44    131        FORMAT (1X,7A4,5X,I2,4X,F9.2,A1,3X,F5.1,5X,F6.1,A1,3X,' N/A ',5X,
          +        F6.1,A1,6X,I1,A1,8X,I1,A1,3X,F7.1,A1)
45    40         CONTINUE
      C
      C   WRITE MISC SYSTEM LINE
      C
46            WGTM = FMCWGT * WEIGHT(NCMPT1)
47            VOLLM = SUM( VOLUME(JMAING),NCMPT1-3)
48            VOLLM = VOLUME(NCMPT1) - VOLLM
49            IF(VOLLM .LE. 0.0) VOLLM = 0.0
50            PCT11 = VOLLM / VOLUME(NCMPT1) * 100.
51            WRITE(7,132) WGTM,FMCWGT,VOLLM,PCT11
52    132     FORMAT(1X,'  9. MISC SYSTEMS/ COMPS     ',5X,' 0',4X,F9.2,1X,
          +  3X,F5.1,5X,F6.1,1X,3X,F5.1,5X,'    0.0',1X,6X,'0',9X,'0',
          +  3X,'      0.0')
      C  OUTPUT SEPARATOR LINE & VEHICLE SUMMARY
53            WRITE (7,135) CV,WEIGHT(NCMPT1),PCWT(NCMPT1),VOLUME(NCMPT1),
          +        PCVCL(NCMPT1),DWNUP(NCMPT1),MATUR(NCMPT1),ICMPLX(NCMPT1),
          +        RELIAB(NCMPT1)
54    135     FORMAT (34X,'__',9X,'____',4X,'_____',6X,'_____',4X,'_____',7X,
          +        '____',6X,'__',8X,'__',6X,'_____'/1X,5A4,14X,'1',4X,F9.2,4X,
          +        F5.1,5X,F6.1,4X,F5.1,5X,F6.1,7X,I1,9X,I1,4X,F7.1)
55            RETURN
56            END
```

OUTPUT ROUTINES - OUTENG SUBROUTINE

```
ISN
 1          SUBROUTINE OLTENG(IV)
       C
       C    ROUTINE TO OUTPUT ENGINEERING PARAMETERS FOR A SPECIFIED VEHICLE
       C
       C  INPUT PARAMETERS:
       C    IV      DESCRIPTION OF VEHICLE OF INTEREST
       C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +     FMACGN(4),10),FMISGN(4),7),IADGUN(4),6),IRANGR(28,7),
          +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +     IELECT(26,6),ICCMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6          COMMON /VPROTO/
 7          REAL*8 FLDNAM,RECNAM,DFAULT
 8          COMMON /JPARS/
 9          COMMON /JPARS1/
```

                    OUTPUT ROUTINES - OUTENG SUBROUTINE

```
ISN
10            CCMMON /NPARS/
11            COMMON /AUX/
12            REAL*8 BL8,DASHES
13            FOUIVALENCE (FNULL,NULL),(IBLS,BLS)
14            COMMON /ARRAYS/
15            REAL*8 MANUF,MODEL,NATION
16            DIMENSION ARRAY(1)
17            EQUIVALENCE (ARRAY(1),DETRNG(1))
18            CCMMON /JPARS2/
        C
19            LOGICAL*1 BL,DF
20            DATA BL/' '/
        C
        C  OUTPUT TABLE TITLE
21            WRITE (7,100)
22      100   FORMAT ('1',' TABLE 2.  CONCEPT VEHICLE, ENGINEERING',
           +      ' PARAMETERS.'/' ')
        C
        C     WRITE FIRST CCMPCNENT NAME
        C
23            WRITE(7,105) (OUTNAM(J,ICMATR),J=1,7)
        C
        C  ITERATE OVER ENGINEERING PAR LABELS
24            LAST = LSTOUT
25            IFRST = IEPARS
26            I = IEPARS
27      2        IF (IOUTF(I) .NE. 0) GO TO 8
        C
        C   TEST IF ANOTHER COMPONENT ON THE VEHICLE
        C
28            IF(I .EQ. IFRST) GO TO 5
29            INST = IVAL(IVFILE(IOFF),NF,JNEXT,IREC)
30               IF(INST .EQ. NULL) GO TO 5
31               IVECLE(ICMP,IV) = INST
32               I = IFRST
33               GO TO 2
```

```
                       OUTPUT ROUTINES - OUTENG SUBROUTINE
ISN
      C  WRITE SPACER OR HEADEP WITHOUT CORRESPONDING VALUE
 34   5         WRITE (7,105) (OUTNAM(J,I),J=1,7)
 35   105       FORMAT (1X,7A4)
 36          IFRST = I
 37             GO TO 80
      C
      C  GET VALUE ASSOCIATED WITH LABEL
 38   8         IFLDRC = IOUTF(I)
 39             ITYPE = IFLDTP(IFLDRC)
 40             IFLD = IFLDVL(IFLCRC)
 41             IRECPT = IFLCRC - IFLD
 42             IRECTP = ITABLE(IFLDPT,IRECPT,NRECTP,1)
 43             NF = NFLDS(IRECTP)
 44             IOFF = IFILPT(IRECTP)
 45             ICMP = IRECTP - 1
 46             IREC = IVECLE(ICMP,IV)
 47             IF (IREC .EQ. NULL) GO TO 80
 48             DO 10 K=1,NTYPES
 49                IF (ITYPES(K) .EQ. ITYPE) GO TO 15
 50   10          CONTINUE
 51             CALL ERR(10,'OUTENG','VEHIC',FLDNAM(IFLDRC),ITYPE)
 52   15        UNIT = UNITS(IFLDRC)
 53             DF = BL
      C  BRANCH ACCORDING TO CATATYPE
 54             GO TO (20,30,40,50),K
      C
      C  REAL VALUE
 55   20        V = DVAL(VFILE(IOFF),NF,IFLD,IREC,IRECTP,UNIT,DF)
 56             WRITE (7,110) (OUTNAM(J,I),J=1,7),V,DF,UNIT
 57   110       FORMAT (1X,7A4,1X,F10.2,A1,3X,A4)
 58             GO TO 80
      C
      C  INTEGER VALUE
 59   30        IVL = IDVAL(IVFILE(ICFF),NF,IFLD,IREC,IRECTP,UNIT,DF)
 60             WRITE (7,115) (OUTNAM(J,I),J=1,7),IVL,DF,UNIT
 61   115       FORMAT (1X,7A4,3X,I8,A1,3X,A4)
```

OUTPUT ROUTINES - OUTENG SUBROUTINE

```
             GO TO 80
C
C  REAL ARRAY
40           IOFF = IFILPT(IRECTP)
             IVL = IVAL(VFILE(IOFF),NF,IFLD,IREC)
             IF (IVL .EQ. NULL) GO TO 20
C  GET POINTER TO ARRAY DESCRIPTION
             DO 42 IARR=1,NARRS
                IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND. IARDF2(JFLDRC,
     +             IARR) .EQ. IFLD) GO TO 44
42              CONTINUE
C  NUMBER OF ELEMENTS IN ARRAY
44           NV = IARDF2(JARSIZ,IARR)
C  OFFSET IN FILE CONTAINING ARRAYS
             IOFFAR = IARDF2(JARRPT,IARR)
C  GET OFFSET OF ARRAY TO BASE
             IOFF = IOFFAR + NV*(IVL-1)
C  WRITE CONTENTS OF ARRAY
             WRITE(7,105) (OUTNAM(J,I),J=1,7)
             CALL OUTARR(ARRAY(IOFF),IRECTP,IFLD)
             GO TO 80
C
C  CHARACTER VALUE  (SHOULDN'T BE ANY)
50           CONTINUE
80           I = I + 1
          IF( I .LE. LAST) GO TO 2
          RETURN
          END
```

OUTPUT ROUTINES - OUTPER SUBROUTINE

```
ISN
 1          SUBROUTINE OUTPER(IV)
      C
      C     ROUTINE TO OUTPUT PERFORMANCE PARAMETERS FOR A SPECIFIED VEHICLE
      C
      C   INPUT PARAMETERS:
      C     IV      DESCRIPTION OF VEHICLE OF INTEREST
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
     +          FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
     +          ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
     +          IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
     +          IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
     +          IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
     +          IELECT(26,6),ICCMMO(28,9),IFIREX(28,10),IENVIR(26,9),
     +          IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
     +          IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
     +          ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
     +          (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
     +          IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
     +          (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
     +          IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
     +          (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
     +          ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
     +          (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
     +          ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
     +          (COMMO(1),ICCMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
     +          IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
     +          (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6          COMMON /VPROTO/
 7          REAL*8 FLDNAM,RECNAM,DFAULT
 8          COMMON /NPARS/
 9          COMMON /AUX/
```

OUTPUT ROUTINES - OUTPER SUBROUTINE

```
ISN
10            REAL*8 BL8,CASHES
11            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
12            COMMON /ARRAYS/
13            REAL*8 MANUF,MODEL,NATICN
14            DIMENSION ARRAY(1)
15            EQUIVALENCE (ARRAY(1),DETRNG(1))
16            COMMON /JPARS2/
        C
17            LOGICAL*1 BL,DF
18            DATA BL/' '/
        C
19            IRECTP = 1
        C
        C  OUTPUT TABLE TITLE
20            WRITE (7,100)
21      100   FORMAT ('1',' TABLE 2.  CONCEPT VEHICLE, PERFORMANCE',
           +      ' PARAMETERS.'/' ')
        C
        C  ITERATE OVER PERFORMANCE PAR LABELS
22            LAST = ICMATR - 1
23            NF = NFLDS(1)
24            DO 80 I=IPPARS,LAST
25               IF (IOUTF(I) .NE. 0) GO TO 8
        C  WRITE SPACER CR HEADER WITHOUT CORRESPONDING VALUE
26      5        WRITE (7,105) (OUTNAM(J,I),J=1,7)
27      105      FORMAT (1X,7A4)
28               GO TO 80
        C
        C  GET VALUE ASSOCIATEC WITH LABEL
29      8        IFLD = IOUTF(I)
30               ITYPE = IFLDTP(IFLD)
31               DO 10 K=1,NTYPES
32                  IF (ITYPES(K) .EQ. ITYPE) GO TO 15
33      10          CONTINUE
34               CALL EPR(10,'OUTPER','VEHIC',FLDNAM(IFLD),ITYPE)
35      15       UNIT = UNITS(IFLD)
```

OUTPUT ROUTINES - OUTPER SUBROUTINE

```
ISN
36              DF = BL
     C  BRANCH ACCORDING TO CATATYPE
37              GO TO (20,30,40,50),K
     C
     C  REAL VALUE
38   20         V = DVAL(IVECLE,NF,IFLD,IV,1,UNIT,DF)
39              WRITE (7,110) (OUTNAM(J,I),J=1,7),V,DF,UNIT
40   110        FORMAT (1X,7A4,1X,F10.2,A1,3X,A4)
41              GO TO 80
     C
     C  INTEGER VALUE
42   30         IVL = IDVAL(IVECLE,NF,IFLD,IV,1,UNIT,DF)
43              WRITE (7,115) (OUTNAM(J,I),J=1,7),IVL,DF,UNIT
44   115        FORMAT (1X,7A4,3X,I8,A1,3X,A4)
45              GO TO 80
     C
     C  REAL ARRAY
46   40         IOFF = IFILPT(IRECTP)
47              IVL = IVAL(VFILE(IOFF),NF,IFLD,IV)
48              IF (IVL .EQ. NULL) GO TO 20
     C  GET POINTER TO ARRAY DESCRIPTION
49              DO 42 IARR=1,NARRS
50                 IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND. IARDF2(JFLDRC,
        +              IARR) .EQ. IFLD) GO TO 44
51   42            CONTINUE
     C  NUMBER OF ELEMENTS IN ARRAY
52   44         NV = IARDF2(JARSIZ,IARR)
     C  OFFSET IN FILE CONTAINING ARRAYS
53              IOFFAR = IARDF2(JARRPT,IARR)
     C  GET OFFSET OF ARRAY TO BASE
54              IOFF = IOFFAR + NV*(IVL-1)
     C  WRITE CONTENTS OF ARRAY
55           WRITE(7,105) (OUTNAM(J,I),J=1,7)
56              CALL OUTARR(ARRAY(IOFF),IRECTP,IFLD)
57              GO TO 80
     C
```

```
                         OUTPUT ROUTINES - OUTPER SUBROUTINE
ISN
      C  CHARACTER VALUE  (SHOULDN'T BE ANY)
58    50          CONTINUE
59    80          CONTINUE
60             RETURN
61             END
```

CUTPUT ROUTINES - OUTVEH SUBROUTINE

```
ISN
 1           SUBROUTINE OLTVEH
       C
       C     ROUTINE TO CLTPUT BEST CANDICATE VEHICLE
       C
 2           CCMMCN /VEH/
 3           DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +      IRELN(3,50),VFILE(1),IVFILE(1)
 4           EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5           EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
       C
 6           CALL TABVAL(ICANDV)
       C
 7           CALL CUTCOM
       C
 8           CALL OUTPER(ICANDV)
       C
```

```
ISN
 9          CALL OUTENG(ICANDV)
      C
10          RETURN
11          END
```

OUTPLT ROUTINES - TABVAL SUBROUTINE

```
ISN
 1          SUBROUTINE TABVAL(IV)
     C
     C          ROUTINE TO PUT VALUES OF INTEREST INTO ARRAYS PRIOR TO PRINTING.
     C     ALSO CALCULATES SUMS, AVERAGES, & PERCENTS OF APPROPRIATE ITEMS.
     C
     C     INPUT PARAMETERS:
     C          IV        VEHCILE CF INTEREST
     C
 2          COMMON /VFH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
           +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
           +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
           +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
           +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
           +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
           +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6          COMMON /VPROTO/
 7          REAL*8 FLDNAM,RECNAM,DFAULT
 8          COMMON /JPARS/
```

OUTPUT ROUTINES - TABVAL SUBROUTINE

```
ISN
  9          COMMON /JPARS1/
 10          CCMMON /NPARS/
 11          COMMON /OUTPLT/
 12          LOGICAL*1 DF
 13          REAL*8 XNAT,XMANUF,XMODEL
 14          COMMON /AUX/
 15          REAL*8 BL8,DASHES
 16          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 17          COMMON /ARRAYS/
 18          REAL*8 MANUF,MODEL,NATION
 19          DIMENSION ARRAY(1)
 20          EQUIVALENCE (ARRAY(1),DETRNG(1))
 21          COMMCN /JPARS2/
 22          COMMON /DATA/
 23          COMMON /DATA2/
       C
       C  INITIALIZE ARRAYS
 24          CALL IZERO(NUMBER,NCMPT1*12)
 25          CALL INITVL(CF,2*NCMPTP,'     ')
 26          CALL INITVL(XMANUF,64,BLS)
 27          CALL INITVL(XMODEL,64,BLS)
 28          CALL INITVL(XNAT,64,BLS)
       C
       C  ITERATE OVER CCMPONENT TYPES
 29          DO 100 ICMP=1,NCMPTP
       C
       C  GET INFORMATION ABOUT THIS COMPONENT TYPE
 30          IRECTP = ICMP + 1
 31          IOFF = IFILPT(IRECTP)
 32          NF = NFLDS(IRECTP)
       C
       C  ITERATE OVER COMPONENTS OF THIS VEHCILE
 33          IREC = IVECLE(ICMP,IV)
 34    10    IF (IREC .EQ. NULL) GO TO 20
       C
       C  INCREMENT NUMBER OF COMPONENTS OF THIS TYPE
```

```
ISN
35           NUMBER(ICMP) = NUMBER(ICMP) + 1
      C
      C  INCREMENT WEIGHT OF COMPONENTS OF THIS TYPE
36           WEIGHT(ICMP) = WEIGHT(ICMP) + DVAL(VFILE(IOFF),NF,JWT,IREC,
          +       IRECTP,'LB  ',DF(3,ICMP))
      C
      C  INCREMENT INTERNAL VOLUME CF COMPONENTS OF THIS TYPE
      C    TEST LOCATION OF COMPCNTNT
      C
37           ICLOC = IVAL(IVFILE(IOFF),NF,JLOC,IREC)
38           IF( ICLOC .EQ. 3) GO TO 12
39           VOLUME(ICMP) = VOLUME(ICMP) + DVAL(VFILE(IOFF),NF,JVOL,IREC,
          +       IRECTP,'FT3 ',DF(4,ICMP))
      C
      C  INCREMENT MAINTENANCE / OPERATIONAL HRS
40       12 DWNUP(ICMP) = DWNUP(ICMP) + DVAL(VFILE(IOFF),NF,JDWNUP,IREC,
          +       IRECTP,'-   ',DF(5,ICMP))
      C
      C  INCREMENT MATURITY INDEX
41           MATUR(ICMP) = MATUR(ICMP) + IDVAL(VFILE(IOFF),NF,JMATUR,IREC,
          +       IRECTP,'-   ',DF(6,ICMP))
      C
      C  INCREMENT COMPLEXITY INDEX
42           ICMPLX(ICMP) = ICMPLX(ICMP) + IDVAL(VFILE(IOFF),NF,JCMPLX,IREC,
          +       IRECTP,'-   ',DF(7,ICMP))
      C
      C  INCREMENT RELIABILITY INDEX
43           RELIAB(ICMP) = RELIAB(ICMP) + DVAL(VFILE(IOFF),NF,JRELIB,IREC,
          +       IRECTP,'-   ',DF(8,ICMP))
      C
      C  DO FOLLOWING OPERATICNS CNLY FOR FIRST COMPONENT OF THIS TYPE
      C
44           IF (NUMBER(ICMP) .NE. 1) GO TO 15
      C
      C  NUMBER OF ITEMS PRODUCED
45           NPROD(ICMP) = IDVAL(VFILE(IOFF),NF,JNUM,IREC,
ISN
             +     IRECTP,'-    ',DF(1,ICMP))
       C
       C  R&D TIME
 46           RDTIME(ICMP) = DVAL(VFILE(IOFF),NF,JRTIME,IREC,
             +     IRECTP,'YR   ',DF(2,ICMP))
       C
       C  NOTE COMPONENT INDEX
 47           INDX(ICMP) = IVAL(VFILE(IOFF),NF,JID,IREC)
       C
       C  NATION
 48           K = IVAL(VFILE(IOFF),NF,JNAT,IREC)
 49           XNAT(ICMP) = NATICN(K)
       C
       C  MANUFACTURER
 50           K = IVAL(VFILE(ICFF),NF,JMANUF,IREC)
 51           XMANUF(ICMP) = MANUF(K)
       C
       C  MODEL
 52           K = IVAL(VFILE(ICFF),NF,JMODEL,IREC)
 53           XMODEL(ICMP) = MODEL(K)
       C
       C  GET NEXT COMPCNENT CF THIS TYPE
 54    15     IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
 55           GO TO 10
       C
       C  IF THERE IS MORE THAN ONE COMPONENT OF THIS TYPE, COMPUTE
       C  AVERAGES FOR CERTAIN OF THE ATTRIBUTES
       C
 56    20     IF (NUMBER(ICMP) .LE. 1) GO TO 100
       C
 57           X = NUMBER(ICMP)
       C
       C  AVERAGE CCMPLEXITY INDEX
 58           ICMPLX(ICMP) = (ICMPLX(ICMP) / X) + 0.5
       C
       C  AVERAGE MATURITY INCEX
```

OUTPUT ROUTINES - TABVAL SUBROUTINE

```
I SN
59          MATUR(ICMP) = (MATUR(ICMP) / X) + 0.5
      C
60    100   CONTINUE
      C
      C  CCMPUTE MEASURES RELEVANT TO VEHICLE AS A WHOLE
      C
61          WEIGHT(NCMPT1) = SUM(WEIGHT,NCMPTP)
62          WEIGHT(NCMPT1) = (1. + FMCWGT) * WEIGHT(NCMPT1)
63          VOLUME(NCMPT1)=VOLUME(JHULL)+VOLUME(JTURET)
64          CALL ARMULT(PCWT,WEIGHT,NCMPT1,100./WEIGHT(NCMPT1))
65          CALL ARMULT(PCVCL(3),VCLUME(3),NCMPT1-2,100./VOLUME(NCMPT1))
66          S = SUM(CMPWTS,NCMPTP)
67          MATUR(NCMPT1) = 0.5 + SUMPD2(MATUR,CMPWTS,NCMPTP) / S
68          ICMPLX(NCMPT1) = 0.5 + SUMPD2(ICMPLX,CMPWTS,NCMPTP) / S
69          DWNUP(NCMPT1) = SUM(DWNUP,NCMPTP)
70          RELIAB(NCMPT1) = 0.0
71          DO 160 ICM=1,NCMPTP
72          IF(RELIAB(ICM) .EQ. 0.0) GO TO 160
73          RELIAB(NCMPT1) = RELIAB(NCMPT1) + 1./RELIAB(ICM)
74      160 CONTINUE
75          IF(RELIAB(NCMPT1) .NE. 0.0) RELIAB(NCMPT1)=1./RELIAB(NCMPT1)
      C
76          VEHCLE(JVWT,IV) = WEIGHT(NCMPT1)
77          VEHCLE(JVVOL,IV) = VOLUME(NCMPT1)
78          RETURN
79          END
```

C.6: UTILITY ROUTINES

This section contains listings of utility routines used for a variety of purposes. Comment cards in the listing of each routine indicate its purpose.

UTILITY ROUTINES - ARMULT SUBROUTINE

```
ISN
 1              SUBROUTINE ARMULT(Y,X,N,W)
      C
      C            ROUTINE TO MULTIPLY EACH ELEMENT OF A VECTOR BY A CONSTANT,
      C         STORING THE RESULT OF EACH MULTIPLICATION IN THE CELLS OF A
      C         VECTOR PARALLEL TO THE FIRST.
      C
      C         INPUT PARAMETERS:
      C            Y        RESULTANT VECTOR
      C            X        INITIAL VECTOR
      C            N        NUMBER OF CELLS IN X & Y
      C            W        THE CONSTANT THAT EACH ELEMENT OF X IS TO MULTIPLIED BY
      C
 2              DIMENSION X(N),Y(N)
      C
 3              DO 10 I=1,N
 4    10           Y(I) = X(I) * W
 5              RETURN
 6              END
```

UTILITY ROUTINES - COPY SUBROUTINE

```
ISN
 1            SUBROUTINE CCPY(X,Y,N)
        C
        C     ROUTINE TO COPY VECTOR X TO VECTOR Y.
        C
        C  INPUT PARAMETERS:
        C     X          VECTOR TO BE COPIED
        C     Y          VECTOR TO RECEIVE CONTENTS OF X
        C     N          SIZE OF VECTORS X AND Y
        C
 2            DIMENSION X(N),Y(N)
        C
 3            DO 10 I=1,N
 4      10       Y(I) = X(I)
 5            RETURN
 6            END
```

UTILITY ROUTINES - DVAL FUNCTION

```
ISN
 1          FUNCTION DVAL(FILE,NFLD,IFLD,IREC,IRECTP,UNIT,DF)
      C
      C        FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
      C     IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT.  AN
      C     INPUT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
      C     DESIRED.  IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
      C     TO THEM.
      C
      C     INPUT PARAMETERS:
      C        FILE     FILE IN WHICH VALUE IS LOCATED
      C        NFLD     NUMBER OF FIELDS IN EACH RECORD OF FILE
      C        IFLD     FIELD OF INTEREST
      C        IREC     RECORD OF INTEREST
      C        IRECTP   RECORD TYPE
      C        UNIT     UNIT OF MEASURE DESIRED
      C
      C     OUTPUT PARAMETERS:
      C        DF       SET TO "*" IF DEFAULT VALUE IS USED
      C
 2          COMMON /VPROTO/
 3          REAL*8 FLDNAM,RECNAM,DFAULT
 4          COMMON /NPARS/
 5          COMMON /AUX/
 6          REAL*8 BL8,DASHES
 7          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
 8          LOGICAL*1 STAR,DF
 9          DIMENSION FILE(NFLD,1)
10          DATA DASH/'-'/,STAR/'*'/
      C
11          DVAL = FILE(IFLD,IREC)
12          IFLDRC = IFLDPT(IRECTP) + IFLD
13          IF (DVAL .NE. FNULL) GO TO 5
14             DVAL = DFAULT(IFLDRC)
15             DF = STAR
```

UTILITY ROUTINES - DVAL FUNCTION

```
ISN
16     5      IF (UNIT .EQ. DASH .OR. UNIT .EQ. UNITS(IFLDRC) .OR. UNITS(IFLDRC)
          +      .EQ. CASH) RETURN
       C   SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
17            DO 10 I=1,NUTRNS
18               IF (UNIT .NE. TUNITS(1,I) .OR. UNITS(IFLDRC) .NE. TUNITS(2,I))
          +        GO TO 8
19               DVAL = DVAL * TUNITS(3,I)
20               RETURN
21     8         IF (UNIT .NE. TUNITS(2,I) .OR. UNITS(IFLDRC) .NE. TUNITS(1,I))
          +        GO TO 10
22               DVAL = DVAL / TUNITS(3,I)
23               RETURN
24     10        CONTINUE
       C
       C   WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
25            CALL ERR(37,'DVAL   ',IRECTP,UNIT,UNITS(IFLDRC))
26            RETURN
27            END
```

UTILITY POUTINES - DVAL2 FUNCTION

```
ISN
 1          FUNCTION DVAL2(X,ICMP,IFLD,UNIT)
      C
      C     FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
      C  IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED F3R IT.  AN
      C  INPUT TO THE FUNCTICN ALSC INDICATES WHAT UNITS OF MEASURE ARE
      C  DESIRED.  IF THE VALUE IS NOT ALRFADY IN THESE UNITS IT IS CONVERTED
      C  TO THEM.
      C
      C  INPUT PARAMETERS:
      C     X        VALUE CF FIFLD
      C     ICMP     COMPCNENT TYPE OF RECORD
      C     IFLD     FIELC OF RECORD OF INTEREST
      C     UNIT     UNIT OF MEASURE DESIRED
      C
 2          COMMON /VPROTO/
 3          REAL*8 FLDNAM,RECNAM,DFAULT
 4          COMMON /NPARS/
 5          CCMMON /AUX/
 6          REAL*8 BL8,DASHES
 7          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
 8          DATA DASH /'-'/
      C
 9          DVAL2 = X
10          IFLDRC = IFLCPT(ICMP+1) + IFLD
11          IF (DVAL2 .EQ. FNULL) DVAL2 = DFAULT(IFLDRC)
12          IF (UNIT .EQ. CASH .OR. UNIT .EQ. UNITS(IFLDRC) .3R. UNITS(IFLDRC)
           +      .FQ. DASH) RETURN
      C  SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
13          DO 10 I=1,NUTRNS
14             IF (UNIT .NE. TUNITS(1,I) .OR. UNITS(IFLDRC) .NE. TUNITS(2,I))
           +      GO TO 8
15             DVAL2 = DVAL2 * TUNITS(3,I)
16             RETURN
17    8        IF (UNIT .NE. TUNITS(2,I) .OR. UNITS(IFLDRC) .NE. TUNITS(1,I))
```

```
ISN
           +      GO TO 10
18               DVAL2 = DVAL2 / TUNITS(3,I)
19               RETURN
20    10         CONTINUE
      C
      C  WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
21             CALL ERR(37,'DVAL2 ',ICMP+1,UNIT,UNITS(IFLDRC))
22             RETURN
23             END
```

UTILITY ROUTINES - ERR SUBROUTINE

```
ISN
 1          SUBROUTINE ERR(IERR,SUBR,PAR1,PAR2,PAR3)
      C
      C         ROUTINE TO PROCESS ERRORS AND WRITE OUT ERROR COMMENTS.
      C     ALSO HAS AN ENTRY "WARN" TO PRINT OUT WARNING MESSAGES.
      C
      C     INPUT PARAMETERS:
      C         IERR    POSITIVE INTEGER IDENTIFYING THE ERROR CONDITION
      C         SUBR    SUBROUTINE IN WHICH THE ERROR OCCURRED
      C         PAR1    THE FIRST PARAMETER OF THE ERROR CONDITION
      C         PAR2    THE SECOND PARAMETER OF THE ERROR CONDITION
      C         PAR3    THE THIRD PARAMETER OF THE ERROR CONDITION
      C
 2          COMMON /VFH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
         +       FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
         +       ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
         +       IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
         +       IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
         +       IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
         +       IELECT(26,6),ICCMMO(28,9),IFIREX(28,10),IENVIR(26,9),
         +       IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
         +       IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
         +       ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
         +       (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
         +       IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
         +       (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
         +       IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
         +       (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
         +       ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
         +       (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
         +       ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
         +       (COMMC(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
         +       IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
         +       (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
```

```
                   UTILITY ROUTINES - ERR SUBROUTINE
ISN
  5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
  6          COMMON /VPROTO/
  7          REAL*8 FLDNAM,RECNAM,DFAULT
  8          COMMON /JPARS/
  9          COMMON /JPARS1/
 10          COMMON /NPARS/
 11          COMMON /OUTPUT/
 12          LOGICAL*1 DF
 13          REAL*8 XNAT,XMANUF,XMODEL
 14          COMMON /AUX/
 15          REAL*8 BL8,CASHES
 16          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
 17          COMMON /ARRAYS/
 18          REAL*8 MANUF,MODEL,NATION
 19          DIMENSION ARRAY(1)
 20          EQUIVALENCE (ARRAY(1),DETRNG(1))
 21          COMMON /JPARS2/
 22          COMMON /SPECS/
 23          REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
 24          COMMON /SPECS2/
 25          REAL*8 VEHS
 26          COMMON /CATA/
 27          COMMON /DATA2/
       C
 28          REAL*8 SUBR
 29          DIMENSION PAR1(2),PAR2(2)
 30          EQUIVALENCE (XPAR1,IPAR1),(XPAR2,IPAR2),(XPAR3,IPAR3)
       C  EQUATE ARGS SO THAT INTEGER PARAMETERS ARE DEFINED.
 31          XPAR1 = PAR1(1)
 32          XPAR2 = PAR2(1)
 33          XPAR3 = PAR3
       C
 34          WRITE (6,2) IERR,SUBR
 35     2    FORMAT ('0 ***  ERROR ',I2,' DETECTED IN SUBROUTINE ',A6,'.')
       C
       C  BRANCH TO APPROPRIATE PIECE OF CODE TO HANDLE ERROR.
```

UTILITY ROUTINES - ERR SUBROUTINE

```
ISN
      C
36           GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,
          +       170,180,190,200,210,220,230,240,250,260,270,280,290,300,
          +       310,320,330,340,350,360,370,380,390,400,410,420,430,440,450),
          +       IERR
      C
37    10     WRITE (6,15) XPAR1,XPAR2
38    15     FORMAT (' THE FIRST NON-COMMENT RECORD OF THE PROTOTYPE FILE'/
          +       ' WAS NOT A VALID VEHICLE DEFINITION RECORD.  IT CONTAINED',
          +       A4/'IN COLUMNS 2-5, AND ',A4,' IN COLUMNS 12-15.')
39           STOP 1
      C
40    20     WRITE (6,25) PAR1,IPAR3,IPAR2
41    25     FORMAT (' THE ATTRIBUTE ',2A4,' OF RECORD TYPE ',I2,' SPECIFIED'/
          +        ' IN THE VEHICLE PROTOTYPE FILE HAS AN OFFSET WITHIN ITS'/
          +        ' RECCRC OF ',I4,', WHICH EXCEEDS THE NUMBER OF FIELDS'/
          +        ' DEFINED FOR THIS RECORD TYPE.')
42           PAUSE
43           GO TO 9999
      C
44    30     WRITE (6,35) IPAR2,IPAR3,PAR1
45    35     FORMAT (' THERE WERE ',I3,' RECORD TYPES ENCOUNTERED IN THE'/
          +        ' VEHICLE PROTOTYPE FILE AND ONLY ',I3,' DEFINED.'/
          +        ' THE LATEST RECODE READ CONTAINS ',2A4,' IN COLS 2-9.')
46           PAUSE
47           GO TO 9999
      C
48    40     WRITE (6,45) PAR1,PAR2
49    45     FORMAT (' THE ALTERNATIVE COMPONENT FILE SPECIFIED THE VEHICLE'/
          +      'CATEGORY ',2A4,', WHICH DOES NOT MATCH THE VEHICLE CATEGORY'/
          +       ' SPECIFIED BY THE PROTOTYPE FILE: ',2A4)
50           PAUSE
51           GO TO 9999
      C
52    50     WRITE (6,55) PAR1,IPAR3
53    55     FORMAT (' A HEADER WAS ENCOUNTERED IN THE ALTERNATIVE COMPONENT'/
ISN
             +     ' FILE IN WHICH THE COMPONENTS WERE NOT SEQUENTIALLY'/
             +     ' NUMBERED.  THE OFFENDING COMPONENT TYPE IS ',2A4,'. THE'/
             +     ' OFFENDING COMPONENT SEQUENCE NUMBER IS ',I3)
 54             PAUSE
 55             GO TO 9999
        C
 56     60      WRITE (6,65) PAR2,PAR1
 57     65      FORMAT (' THE ATTRIBUTE ',2A4,' APPEARING UNDER COMPONENT ',2A4/
             +     ' IN THE ALTERNATIVE VEHCILE FILE COULD NOT BE FOUND IN THE'/
             +     ' LIST OF LEGAL ATTRIBUTES FOR THIS COMPONENT.')
 58             PAUSE
 59             GO TO 9999
        C
 60     70      WRITE (6,75) IPAR1
 61     75      FORMAT (' THE RECORD TYPE SPECIFIED, ',I3,', WAS OUT OF BOUNDS.')
 62             PAUSE
 63             GO TO 9999
        C
 64     80      WRITE (6,85) IPAR1,IPAR2
 65     85      FORMAT (' HAVE RUN OUT OF RECORDS OF TYPE ',I3,'.'/
             +     ' MORE THAT ',I4,' RECORDS OF THIS TYPE WERE NEEDED.'/
             +     ' NEED TO REDIMENSION APPROPRIATE ARRAYS, CHANGE BLOCKDATA,'/
             +     ' AND RECOMPILE.')
 66             PAUSE
 67             GO TO 9999
        C
 68     90      WRITE (6,95) PAR1
 69     95      FORMAT (' THE HEADER ',2A4,' READ FROM THE ALTERNATIVE COMPONENT'/
             +     ' FILE WAS NOT RECOGNIZED AS A LEGAL COMPONENT.')
 70             PAUSE
 71             GO TO 9999
        C
 72     100     WRITE (6,105) XPAR3,PAR2,PAR1
 73     105     FORMAT (' THE DATATYPE ',A4,' STORED FOR ATTRIBUTE ',2A4,' OF'/
             +     ' COMPONENT ',2A4,' DOES NOT MATCH ANY OF THE LEGAL'/
             +     ' ALTERNATIVES.')
```

```
ISN
74              PAUSE
75              GO TO 9999
      C
76    110       WRITE (6,115) XPAR1,XPAR2
77    115       FORMAT (' THE IDENTIFYING HEADER ',A4,' READ FROM THE USER '/
             +       ' SPECIFICATION FILE DID NOT MATCH THAT EXPECTED: ',A4)
78              PAUSE
79              GO TO 9999
      C
80    120       WRITE (6,125) PAR1,PAR2
81    125       FORMAT (' THE USER SPECIFICATION FILE DESIGNATED A VEHICLE'/
             +       ' CATEGORY ',2A4,' WHICH DOES NOT MATCH THE VEHICLE CATEGORY'/
             +       ' SPECIFIED BY THE PROTOTYPE FILE: ',2A4)
82              PAUSE
83              GO TO 9999
      C
84    130       WRITE (6,135) PAR1,IPAR2
85    135       FORMAT (' AN INSUFFICIENT NUMBER OF ',2A4,' RECORDS, ',I3,', '/
             +       ' WERE SET ASIDE TO STORE THE USERS SPECS.  ARRAYS'/
             +       ' DIMENSICNED ON THIS SUBSCRIPT SHOULD BE ENLARGED, BLOCK'/
             +       ' DATA CHANGED, AND THE PROGRAM RECOMPILED.')
86              PAUSE
87              GO TO 9999
      C
88    140       WRITE (6,145) XPAR1
89    145       FORMAT (' THE FIRST CHARACTER OF A HEADER RECORD FOR'/
             +   ' (1) COMPONENTS, (2) ENGINEERING PARAMETERS, OR (3) (VEHICLE)'/
             +       'PERFORMANCE PARAMETERS WAS "',A1,'", NOT "C", "E",'/
             +       '"P", OR "V",')
90              PAUSE
91              GO TO 9999
      C
92    150       WRITE (6,155) XPAR1
93    155       FORMAT (' THE TYPE FIELD OF A RECORD DESCRIBING A'/
             +       ' VEHICLE PERFORMANCE PARAMETER CONSTRAINT CONTAINED'/
             +       '"',A3,'", NOT "    " OR "VEH".')
```

```
 ISN
  94          PAUSE
  95          GO TO 9999
        C
  96    160   WRITE (6,165) IPAR2,IPAR1
  97    165   FORMAT (' THE PROTOTYPE FILE SPECIFIED THAT FIELD ', I3/
             +      ' OF RECORD TYPE ',I3,' HAD A DATATYPE "RA".'/
             +      ' THIS DOES NOT MATCH WITH THE ARRAY IARDF2 STORING'/
             +      ' INFORMATICN ABOUT FIELDS WHICH MAY HAVE VALUES'/
             +      ' WHICH ARE ARRAYS.')
  98          PAUSE
  99          GO TO 9999
        C
 100    170   WRITE (6,175) IPAR3,PAR1,IPAR2
 101    175   FORMAT (' HAVE RUN OUT OF SPACE IN STORING ARRAY TYPE ',I2/
             +      ' AS A VALUE OF ATTRIBUTE ',2A4,'.  NEED TO ENLARGE'/
             +      ' NUMBER OF SUCH ARRAYS BEYOND THE ',I4, ' CJRRENTLY'/
             +      ' ALLOCATED.  CHANGE AFFECTED ARRAY SIZES & BLOCKDATA'/
             +      ' AND RECOMPILE.')
 102          PAUSE
 103          GO TO 9999
        C
 104    180   WRITE (6,185) PAR2,PAR1
 105    185   FORMAT (' THE USER SPECIFICATION FILE DESIGNATED A VEHICLE '/
             +    ' CATEGORY ',2A4,' WHICH DOES NOT MATCH THE VEHICLE CATEGORY'/
             +      ' SPECIFICIFIED BY THE EXISTING VEHICLE FILE: ',2A4)
 106          PAUSE
 107          GO TO 9999
        C
 108    190   WRITE (6,195) IPAR1
 109    195   FORMAT (' USER COMPONENT SPECIFICATION # ',I3,' CALLED FOR THE'/
             +      ' USE OF AN EXISTING VEHICLE BY HAVING "VEH" IN COLUMNS 2-4'/
             +      ' BUT FIALED TO INDICATE WHICH EXISTING VEHICLE.')
 110          PAUSE
 111          GO TO 9999
        C
 112    200   WRITE (6,205) PAR1,IPAR2
```

UTILITY ROUTINES - ERR SUBROUTINE

```
 ISN
113    205    FORMAT (' VEHICLE NAME ',2A4,' DID NOT FIT  IN THE VEHS ARRAY.'/
             +      ' MUST EXPAND THE SIZE FO THIS ARRAY BEYOND ',I3)
114           PAUSE
115           GO TO 9999
       C
116    210    WRITE (6,215) IPAR1,IPAR2
117    215    FORMAT (' ARRAY CVERFLOW FOR ARRAY IRECS.  MUST EXPAND ITS SIZE'/
             +       ' BEYOND ',I3,'.  THE VEHICLE NAME BEING PROCESSED WHEN'/
             +       ' OVERFLOW CCCURRED WAS #',I2)
118           PAUSE
119           GO TO 9999
       C
120    22)    WRITE (6,225) XPAR1,PAR2
121    225    FORMAT (' THE COMPONENT CODE "',A2,'" SPECIFIED BY THE '/
             +     ' ENGINEERING PARAMETER SPECIFICATION REFERRING TO ATTRIBUTE',
             +      2A4/' COULD NOT BE LOCATED IN THE LIST OF VALID COMPONENT ',
             +       ' CODES.  THIS SPECIFICATION WILL BE SKIPPED.')
122           RETURN
       C
123    230    WRITE (6,235) IPAR3,IPAR2,IPAR1
124    235    FORMAT (' THE NUMRER OF PARAMETERS CALLED FOR BY JSER '/
             +       ' SPECIFICATION # ',I3,' FOR A COMPONENT OF TYPE ',I2/
             +       ' WAS ',I3,', WHICH IS OUTSIDE THE RANGE DEFINED BY THE'/
             +       ' MINIMLM AND MAXIMUM FOR THIS COMPONENT TYPE SPECIFIED '/
             +       ' IN THE PROTOTYPE FILE.  HOWEVER THE USER SPEC WILL BE'/
             +       ' USED ANYWAY.')
125           RETURN
       C
126    240    WRITE (6,245) IPAR3,IPAR2,IPAR1
127    245    FORMAT (' USER SPECIFICATION # ',I3,' FOR COMPONENT OF TYPE ',A3/
             +     ' REFERRED TO COMPONENT INDEX ',I3,', WHICH IS OUT OF BOUNDS'/
             +       ' OF THE LEGAL RANGE OF SUCH INDECES.')
128           PAUSE
129           GO TO 9999
       C
13)    25)    WRITE (6,255) PAR2,IPAR3,XPAR1
```

UTILITY ROUTINES - ERR SUBROUTINE

```
 ISN
131    255   FORMAT (' THE ATTRIBUTE ',2A4,' SPECIFIED IN THE USER ENGINEERING'/
            +      ' PARAMETER CONSTRAINT # ',I3,', COULD NOT BE FOUND AMONGST'/
            +      ' THE LEGAL LIST OF ATTRIBUTES FOR COMPONENT ',A2,'.  THIS'/
            +      ' SPECIFICATION WILL BE IGNORED.')
132          RETURN
       C
133    260   WRITE (6,265) XPAR3,XPAR1,PAR2
134    265   FORMAT (' THE RELOP "',A4,'" SPECIFIED BY THE USER '/
            +     ' SPECIFICATION REFERRING TO COMPONENT ',A2,' AND ATTRIBUTE ',
            +      2A4/' WAS NOT AMONGST THE LIST OF LEGAL RELATIONAL'/
            +      ' OPERATORS.  THIS SPECIFICATION WILL BE IGNORED.')
135          RETURN
       C
136    270   WRITE (6,275) PAR2,IPAR3,RELNAM(IPAR1)
137    275   FORMAT (' A CONSTRAINT ASSOCIATED WITH ATTRIBUTE ',2A4/
            +      ' OF RECORD TYPE ',I2,' SPECIFIED A "',A2,'" RELATION'/
            +      ' INVOLOVING CHARACTER DATA.')
138          PAUSE
139          GO TO 9999
       C
140    280   WRITE (6,285) PAR1
141    285   FORMAT (' THE COMPONENT NAME ',2A4,' SPECIFIED IN THE EXISTING'/
            +      ' VEHICLE FILE DOSE NOT MATCH ANY OF THE LEGAL COMPONENT'/
            +      ' NAMES STORED.')
142          PAUSE
143          GO TO 9999
       C
144    290   WRITE (6,295) IPAR2,IPAR3,IPAR1
145    295   FORMAT (' A CONSTRAINT IN THE USER SPECIFICATION FILE INDICATED '/
            +     ' THAT FIELD ',I3,' OF RECORD ID ',I3,' HAVING RECORD TYPE',
            +     I4,' WAS TO BE NOT EQUAL A SPECIFIED VALUE.  THIS IS NOT'/
            +      ' A LEGAL CONSTRAINT TO SPECIFY IN THIS SITUATION.')
146          PAUSE
147          GO TO 9999
       C
148    300   WRITE (6,305) IPAR1
```

UTILITY ROUTINES - ERR SUBROUTINE

```
  305   FORMAT (' HAVE RUN OUT OF RELATICN RECORDS.  MUST INCREASE THE'/
      +      ' NUMBER ABOVE ',I3,' AND RECOMPILE.')
        PAUSE
        GO TO 9999
C
  310   WRITE (6,315) IPAR1
  315   FORMAT (' HAVE RUN OUT OF RECORDS IN "OUTNAM".  THE LAST ONE'/
      +    ' USED WAS ',I4,'.  NEED TO INCREASE THE SIZE OF THIS ARRAY.')
        PAUSE
        GO TO 9999
C
  320   WRITE (6,325)
  325   FORMAT (' DATA FOR THE FOLLOWING USER SPECIFICATIONS WERE NOT '/
      +      ' FCUND IN THE EXISTING VEHICLE FILE:'/' '/
      +      '      TYPE SPEC    SPEC      VEHICLE')
        IR = IPAR1
  322   IF (IR .EQ. NULL) GO TO 328
        IV = IRECS(JVEH,IR)
        WRITE (6,326) IRECS(JTYP,IR),IRECS(JSPEC,IR),VEHS(IV)
  326   FORMAT (I6,I8,6X,A8)
        IR = IRECS(JNXT,IR)
        GO TO 322
  328   PAUSE
        GO TO 9999
C
  330   WRITE (6,335) IPAR1
  335   FORMAT (' HAVE ENCOUNTERED A PREMATURE END TO THE EXISTING'/
      +      ' VEHICLE FILE.  LSTVEH = ',I3)
        PAUSE
        GO TO 9999
C
  340   CONTINUE
C
  350   CONTINUE
C
  360   CONTINUE
```

UTILITY ROUTINES - ERR SUBROUTINE

```
 ISN
        C
 174    370   WRITE (6,375) IPAR1,XPAR3,XPAR2
 175    375   FORMAT (' AN ATTEMPT TO TRANSLATE THE VALUE IN A FIELD OF A'/
             +      ' RECORD OF RECORD TYPE ',I3,' FROM ',A4,' TO ',A4,' FAILED'/
             +      ' BECAUSE THIS PAIR OF UNITS WAS NOT FOUND IN "TUNITS".')
 176          PAUSE
 177          GO TO 9999
        C
 178    380   WRITE (6,385) IPAR1
 179    385   FORMAT (' AN OUTPUT ARRAY WAS ENCOUNTERED WITH ',I3,' DIMENSIONS'/
             +      ' BUT THE PROGRAM WAS WRITTEN TO HANDLE ONLY ARRAYS HAVING'/
             +      ' 1 TO 3 DIMENSIONS.')
 180          PAUSE
 181          GO TO 9999
        C
 182    390   CONTINUE
        C
 183    400   WRITE (6,405) XPAR3,XPAR1,PAR2,PAR2
 184    405   FORMAT (' THE DISTANCE (',F8.1,') SPECIFIED BY VARIABLE ',A4/
             +      ' ASSOCIATED WITH THE SELECTED ROAD WHEEL COMPONENT AND'/
             +      ' GIVING THE VERTICAL DISTANCE BETWEEN THE ROAD WHEELS AND '/
             +    A8,', WAS TOO SMALL TO KEEP THE ROAD WHEELS FROM INTERFERING'/
             +      ' WITH THE ',A8)
 185          PAUSE
 186          GO TO 9999
        C
 187    410   WRITE (6,415) IPAR1,MINNUM,MAXNUM
 188    415   FORMAT (' THE NUMBER OF ROAD WHEELS SPECIFIED, ',I2,' WAS NOT'/
             +      ' BETWEEN THE ALLOWED MINIMUM, ',I2,' AND MAXIMUM,',I3)
 189          PAUSE
 190          GO TO 9999
        C
 191    420   WRITE (6,425) IPAR3,IPAR1
 192    425   FORMAT (' A SATISFACTORY CANDIDATE VEHICLE COULD NOT BE FOUND'/
             +      ' BECAUSE NONE OF THE ',I2,' ALTERNATIVES OF COMPONENT TYPE'/
             +      I3,' WERE FOUND TO BE SUITABLE.')
```

UTILITY ROUTINES - ERR SUBROUTINE

```
 ISN
 193          PAUSE
 194          GO TO 9999
       C
 195   430    WRITE (6,435)
 196   435    FORMAT (' NONE OF THE COMBINATIONS OF ALTERNATIVE COMPONENTS'/
            +      ' WERE BOTH CONSISTENT WITH EACH OTHER AND WITH THE '/
            +      ' USERS SPECIFICATIONS.  THEREFORE A CANDIDATE VEHICLE'/
            +      ' SOLUTION WAS NOT FOUND.  THE USER SHOULD RUN THE PROGRAM'/
            +      ' AGAIN, REMOVING SOME CONSTRAINTS.')
 197          PAUSE
 198          GO TO 9999
       C
 199   440    CONTINUE
       C
 200   450    I = IPAR1 - 1
 201          WRITE (6,455) I
 202   455    FORMAT (' THE STABILIZATION PERFORMANCE INDEX SPECIFIED IN THE '/
            +      ' SELECTED "STABLE" COMPONENT RECORD WAS OUT-OF-BOUNDS (NOT'/
            +      ' 1 OR 2).  IT WAS: ',I3)
 203          PAUSE
 204          GO TO 9999
       C
 205          ENTRY WARN(IWARN,SUBR,PAR1,PAR2,PAR3)
       C
       C  EQUATE ARGS SO THAT INTEGER PARAMETERS ARE DEFINED
       C
 206          XPAR1 = PAR1(1)
 207          XPAR2 = PAR2(1)
 208          XPAR3 = PAR3
       C
       C  BRANCH TO APPROPRIATE PIECE OF CODE TO HANDLE WARNING MESSAGE
       C
 209          GO TO (510,520),IWARN
       C
 210   510    ICMP = IPAR1 - 1
 211          WRITE (6,515) IPAR2,IPAR3,ICMP
 ISN
212     515   FORMAT (' A CONSTRAINT SPECIFIED IN THE USER SPECIFICATION FILE'/
            +      ' HAS BEEN USED TO CHANGE THE VALUE OF FIELD ',I3,' OF '/
            +      ' RECCRD ID ',I3,' OF COMPCNENT TYPE ',I3,'.  IF THIS'/
            +      ' MODIFICATION OF THE COMPONENT REQUIRES DEVELOPMENT TIME'/
            +      ' OR COST, CR IF THE COMPONENT HAS BEEN USED TO DEVELOP'/
            +      ' PERFORMANCE DATA FOR THIS RUN, ONE SHOULD TAKE THESE'/
            +      ' FACTORS INTO ACCOUNT IN INTERPRETING THE OUTPUT OF THIS',
            +      ' RUN.')
213           RETURN
        C
214     520   WRITE (6,525)
215     525   FORMAT (' THE GUN DEPRESSION ANGLE SPECIFIED IN THE SELECTED'/
            +      ' MAIN GUN COULD NOT BE ACHIEVED WITH THE HULL & TURRET'/
            +      ' DIMENSIONS GIVEN.  THE MAIN GUNS HEIGHT ABOVE THE ',A8/
            +      ' OF THE HULL WAS THEREFORE INCREASED FROM ',F6.0,' INCHES'/
            +      ' TO ',F6.0,' INCHES.')
216           RETURN
        C
217     9999  RETURN
218           END
```

UTILITY ROUTINES - FVAL1 FUNCTION

```
ISN
 1            FUNCTION FVAL1(X,R,N,V)
       C
       C        FUNCTION TO INTERPOLATE OR EXTRAPOLATE ON AN ARRAY AND RETRIEVE THE
       C      FUNCTIONAL VALUE CORRESPONDING TO AN INDEPENDENT VARIABLE.  THIS
       C      VALUE IS RETURNED BY THE FUNCTION.  THERE ARE TWO OTHER ENTRY POINTS,
       C      FVAL2 & FVAL3, FOR FUNCTIONS OF TWO OR THREE VARIABLES.  THIS
       C      FUNCTION USES A ROUGH APPROXIMATION METHOD OF INTERPOLATION IN THESE
       C      LATTER CASES.
       C
       C      INPUT PARAMETERS:
       C         X         ARRAY STORING THE FUNCTION
       C         R         VECTOR STORING VALUES OF DEPENDENT VARIABLES
       C         N         NUMBER OF VALUES TABULATED IN R & N
       C         V         INDEPENDENT VALUE TO USE IN INTERPOLATION
       C
 2            DIMENSION X(N),R(N)
 3            DIMENSION XX(N1,N2),R1(N1),R2(N2)
 4            DIMENSION XXX(N1,N2,N3),R3(N3)
       C
 5            DO 10 I=2,N
 6               IF (V .GT. R(I) .AND. I .NE. N) GO TO 10
 7               SLOPE = (X(I) - X(I-1)) / (R(I) - R(I-1))
 8               FVAL1 = X(I-1) + (V - R(I-1))*SLOPE
 9               RETURN
10     10        CONTINUE
11            PAUSE
       C
12            ENTRY FVAL2(XX,R1,N1,V1,R2,N2,V2)
       C
13            DO 30 I=2,N1
14               IF (V1 .GT. R1(I) .AND. I .NE. N1) GO TO 30
15               DO 20 J=2,N2
16                  IF (V2 .GT. R2(J) .AND. J .NE. N2) GO TO 20
17                  SLOPE1 = (XX(I,J) - XX(I-1,J)) / (R1(I) - R1(I-1))
18                  SLOPE2 = (XX(I-1,J) - XX(I-1,J-1)) / (R2(J) - R2(J-1))
```

```
                         UTILITY ROUTINES - FVAL1 FUNCTION
ISN
19                 FVAL2 = XX(I-1,J-1) + (V1 - R1(I-1))*SLOPE1
           +          + (V2 - R2(J-1)) * SLOPE2
20                 RETURN
21    20        CONTINUE
22    30     CONTINUE
23           PAUSE
      C
24           ENTRY FVAL3(XXX,R1,N1,V1,R2,N2,V2,R3,N3,V3)
      C
25           DO 60 I=2,N1
26              IF (V1 .GT. R1(I) .AND. I .NE. N1) GO TO 60
27              DO 50 J=2,N2
28                 IF (V2 .GT. R2(J) .AND. J .NE. N2) GO TO 50
29                 DO 40 K=2,N3
30                    IF (V3 .GT. R3(K) .AND. K .NE. N3) GO TO 40
31                    SLOPE1 = (XXX(I,J,K) - XXX(I-1,J,K)) / (R1(I) - R1(I-1))
32                    SLOPE2 = (XXX(I-1,J,K) - XXX(I-1,J-1,K)) /(R2(J)-R2(J-1))
33                    SLOPE3 = (XXX(I-1,J-1,K) - XXX(I-1,J-1,K-1)) /
           +                (R3(K) - R3(K-1))
34                    FVAL3 = XXX(I-1,J-1,K-1) + (V1 - R1(I-1))*SLOPE1
           +              + (V2 - R2(J-1)) * SLOPE2
           +              + (V3 - R3(K-1)) * SLOPE3
35                    RETURN
36    40              CONTINUE
37    50           CONTINUE
38    60        CONTINUE
39           PAUSE
40           END
```

UTILITY ROUTINES - IATTR FUNCTION

```
ISN
 1            FUNCTICN IATTR(ATTR,CMP,IRECTP)
       C
       C         FUNCTION FOR COMPUTING THE INDEX OF AN ATTRIBUTE WITHIN A
       C      COMPONENT PECORC.  FUNCTION RETURNS THIS INDEX AS ITS VALUE OR
       C      RETURNS A NULL VALUE IF THE SPECIFIED ATTRIBUTE NAME DOES NOT
       C      IDENTIFY A LEGAL ATTRIBUTE OF THIS COMPONENT.  FUNCTIJN ALSO
       C      RETURNS THE INDEX OF THE RECORD TYPE CORRESPONDING TO CMP.
       C
       C      INPUT PARAMETERS:
       C         ATTR    ATTRIBUTE NAME
       C         CMP     COMPONENT NAME
       C
       C      OUTPUT PARAMETERS:
       C         IRECTP  RECORD TYPE INDEX OF CMP
       C
       C      ALSO THERE IS AN ENTRY POINT IATTR2 WHERE BOTH ATTR AND IRECTP ARE
       C      INPUT AND THE FIELD INDEX COMPUTED.
       C
 2            COMMON /VPRCTO/
 3            REAL*8 FLDNAM,RECNAM,DFAULT
 4            CCMMON /NPARS/
 5            COMMON /AUX/
 6            REAL*8 BL8,DASHES
 7            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
       C
 8            REAL*8 ATTR,CMP
       C
       C      DETERMINE INDEX OF RECCRD STORING COMPONENT
 9            DO 10 IRECTP=1,NRECTP
10               IF (RECNAM(IRECTP) .EQ. CMP) GO TO 15
11     10        CONTINUE
12            CALL ERR(6,'IATTR ',CMP,ATTR,0)
13            IATTP = NULL
14            IRECTP = NULL
15            RETURN
```

UTILITY ROUTINES - IATTR FUNCTION

```
ISN
      C
16             ENTRY IATTR2(ATTR,IRECTP)
      C
      C  DETERMINE INDEX OF ATTRIBUTE WITHIN COMPONENT RECORD
17    15       J1 = IFLDPT(IRECTP) + 1
18             JN = IFLDPT(IRECTP+1)
19             DO 20 J=J1,JN
20                IF (ATTR .EQ. FLDNAM(J)) GO TO 25
21    20          CONTINUE
22             IATTR = NULL
23             IATTR2 = IATTR
24             RETURN
      C
25    25       IATTR = IFLDVL(J)
26             IATTR2 = IATTR
27             RETURN
28             END
```

```
ISN
 1            FUNCTICN IDVAL(IFILE,NFLD,IFLD,IREC,IRECTP,UNIT,DF)
        C
        C        FUNCTICN TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
        C     IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT.  AN
        C     INPØT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
        C     DESIRED.  IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
        C     TO THEM.
        C
        C     INPUT PARAMETERS:
        C        IFILE    FILE IN WHICH VALUE IS LOCATED
        C        NFLD     NUMBER OF FIELDS IN EACH RECORD OF FILE
        C        IFLD     FIELC OF INTEREST
        C        IREC     RECORD OF INTEREST
        C        IRECTP   RECORD TYPE
        C        UNIT     UNIT OF MEASURE DESIRED
        C
        C     OUTPUT PARAMETERS:
        C        DF       SET TO "*" IF DEFAULT VALUE IS USED
        C
 2            COMMON /VPROTO/
 3            REAL*8 FLDNAM,RECNAM,DFAULT
 4            COMMON /NPARS/
 5            CCMMON /AUX/
 6            REAL*8 BL8,DASHES
 7            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
        C
 8            LOGICAL*1 STAR,DF
 9            DIMENSION IFILE(NFLD,1)
10            DATA CASH/'-'/,STAR/'*'/
        C
11            IDVAL = IFILE(IFLD,IREC)
12            IFLDRC = IFLCPT(IRECTP) + IFLD
13            IF (IDVAL .NE. NULL) GO TO 5
14               IDVAL = DFAULT(IFLDRC)
15               DF = STAR
```

UTILITY ROUTINES - IDVAL FUNCTION

```
    5     IF (UNIT .EQ. DASH .OR. UNIT .EQ. UNITS(IFLDRC) .OR. UNITS(IFLDRC)
         +     .EQ. DASH) RETURN
    C  SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
          DO 10 I=1,NUTRNS
             IF (UNIT .NE. TUNITS(1,I) .OR. UNITS(IFLDRC) .NE. TUNITS(2,I))
         +      GO TO 8
             IDVAL = IDVAL * TUNITS(3,I)
             RETURN
    8        IF (UNIT .NE. TUNITS(2,I) .OR. UNITS(IFLDRC) .NE. TUNITS(1,I))
         +      GO TO 10
             IDVAL = IDVAL / TUNITS(3,I)
             RETURN
    10       CONTINUE
    C
    C  WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
          CALL ERR(37,'IDVAL ',IRECTP,UNIT,UNITS(IFLDRC))
          RETURN
          END
```

```
                        UTILITY ROUTINES - IDVAL2 FUNCTION
ISN
 1             FUNCTION IDVAL2(IX,ICMP,IFLD,UNIT)
      C
      C        FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
      C     IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT.  AN
      C     INPUT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
      C     DESIRED.  IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
      C     TO THEM.
      C
      C     INPUT PARAMETERS:
      C        X         VALUE OF FIELD
      C        ICMP      COMPONENT TYPE OF RECORD
      C        IFLD      FIELD OF RECORD OF INTEREST
      C        UNIT      UNIT OF MEASURE DESIRED
      C
 2             COMMON /VPROTO/
 3             REAL*8 FLDNAM,RECNAM,DFAULT
 4             COMMON /NPARS/
 5             COMMON /AUX/
 6             REAL*8 BL8,DASHES
 7             EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
 8             DATA DASH /'-'/
      C
 9             IDVAL2 = IX
10             IFLDRC = IFLDPT(ICMP+1) + IFLD
11             IF (IDVAL2 .EQ. NULL) IDVAL2 = DFAULT(IFLDRC)
12             IF (UNIT .EQ. DASH .OR. UNIT .EQ. UNITS(IFLDRC) .OR. UNITS(IFLDRC)
           +      .EQ. DASH) RETURN
      C     SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
13             DO 10 I=1,NUTRNS
14                IF (UNIT .NE. TUNITS(1,I) .OR. UNITS(IFLDRC) .NE. TUNITS(2,I))
           +        GO TO 8
15                IDVAL2 = IDVAL2 * TUNITS(3,I)
16                RETURN
17    8           IF (UNIT .NE. TUNITS(2,I) .OR. UNITS(IFLDRC) .NE. TUNITS(1,I))
```

```
                        UTILITY ROUTINES - IDVAL2 FUNCTION
ISN
              +     GO TO 10
18                 IDVAL2 = IDVAL2 / TUNITS(3,I)
19                 RETURN
20     10          CONTINUE
       C
       C  WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
21               CALL ERR(37,'IDVAL2',ICMP+1,UNIT,UNITS(IFLDRC))
22               RETURN
23               END
```

UTILITY ROUTINES - INITVL SUBROUTINE

```
ISN
 1          SUBROUTINE INITVL(IVEC,N,IVAL)
      C
      C     SUBROUTINE TO INITIALIZE ALL THE ELEMENTS OF AN INTEGER ARRAY TO
      C  A SPECIFIED VALUE.
      C
      C  INPUT PARAMTERS:
      C     IVEC      A BLOCK OF CONTIGUOUS STORAGE TO BE INITIALIZED
      C               (IT NEED NOT BE A SINGLE 1-DIM ARRAY IN THE CALLING
      C               ROUTINE, THOUGH IT IS TREATED AS SUCH IN INITVL).
      C     N         THE NUMBER OF COMPUTER WORDS IN IVEC TO BE INITIALIZED.
      C     IVAL      THE VALUE TO SET EACH OF THE WORDS OF IVEC TO.
      C
 2          DIMENSION IVEC(N)
 3          DO 10 I=1,N
 4    10        IVEC(I) = IVAL
 5          RETURN
 6          END
```

UTILITY ROUTINES - INLIST FUNCTION

```
ISN
 1            FUNCTION INLIST(IVAL,IFILE,NF,IFLD,JNEXT,IREC)
        C
        C     ROUTINE SEARCHES A LINKED LIST OF RECORDS FOR A RECORD WITH A
        C  SPECIFIED VALUE IN A SPECIFIED FIELD.  FUNCTION RETURNS A POINTER
        C  TO THE RECORD IF IT FINDS ONE, NULL OTHERWISE.
        C
        C  INPUT PARAMETERS:
        C     IVAL     VALUE TO LOOK  FOR
        C     IFILE    FILE IN WHICH RECORDS CHAINED TOGETHER  ARE LOCATED
        C     NF       NUMBER OF FIELDS IN EACH RECORD
        C     IFLD     FIELD IN WHICH TO SEARCH FOR VALUE
        C     JNEXT    FIELD USED FOR CHAINING RECORDS
        C     IREC     FIRST RECORD IN LIST
        C
 2            COMMON /AUX/
 3            REAL*8 BL8,DASHES
 4            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
        C
 5            DIMENSION IFILE(NF,1)
        C
 6            INLIST = IREC
 7      5     IF (INLIST .EQ. NULL) RETURN
 8            IF (IFILE(IFLD,INLIST) .EQ. IVAL) RETURN
 9            INLIST = IFILE(JNEXT,INLIST)
10            GO TO 5
11            END
```

UTILITY ROUTINES - FUNCTION IOKVAL

```
ISN
 1          LOGICAL FUNCTION IOKVAL(IVALUE,IRECTP,IFLD,ISAV)
      C
      C     FUNCTION TO TEST WHETHER THE USER HAS SPECIFIED ANY CONSTRAINTS
      C  ON A PARTICULAR ATTRIBUTE'S VALUE AND IF SO WHETHER THE SPECIFIED
      C  VALUE SATISFIES THESE CONSTRAINTS.  THE FUNCTION RETURNS .TRUE. IF NO
      C  CONSTRAINTS ARE SPECIFIED OR IF THE INDICATED VALUE SATISFIES ALL
      C  CONSTRAINTS.
      C
      C  INPUT PARAMETERS:
      C     IVALUE  VALUE TO BE TESTED
      C     IRECTP  RECORD TYPE IN WHICH VALUE IS LOCATED
      C     IFLD    FIELD OF RECORD IN WHICH VALUE LOCATED
      C    ISAV     RECORD NUMBER WHICH RELATION REFERENCE STORED
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
           +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
           +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
           +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
           +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
ISN
          +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
  5             EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
  6             COMMON /VPROTO/
  7             REAL*8 FLDNAM,RECNAM,DFAULT
  8             COMMON /JPARS/
  9             COMMON /JPARS1/
 10             COMMON /AUX/
 11             REAL*8 BL8,DASHES
 12             EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
        C
 13             IOKVAL = .TRUE.
        C
        C  GET CONSTRAINT RECORD IF ONE EXISTS
 14             NF = NFLDS(IRECTP)
 15             IOFF = IFILPT(IRECTP)
        C
        C   TEST IF RELATICN REFERENCE PASSED
        C
 16             IF(ISAV .EQ. NULL) GO TO 3
 17             IREC=ISAV
 18             GO TO 4
 19     3       IREC = ISPECV
 20             IF (IRECTP .GT. 1) IREC = IVECLE(IRECTP-1,ISPECV)
 21             IF (IREC .EQ. NULL) RETURN
 22     4       IR = IVAL(VFILE(IOFF),NF,IFLD,IREC)
 23     5       IF (IR .EQ. NULL) RETURN
 24             IV = IRELN(JVALUE,IR)
 25             IRTYP = IRELN(JRELOP,IR)
 26             GO TO (10,15,20,25,30,35),IRTYP
 27     10      IF (IVALUE .LT. IV) GO TO 40
 28                IOKVAL = .FALSE.
 29                RETURN
 30     15      IF (IVALUE .LE. IV) GO TO 40
 31                IOKVAL = .FALSE.
 32                RETURN
```

```
ISN
33    20     IF (IVALUE .EQ. IV) GO TO 40
34              IOKVAL = .FALSE.
35              RETURN
36    25     IF (IVALUE .NE. IV) GO TO 40
37              IOKVAL = .FALSE.
38              RETURN
39    30     IF (IVALUE .GE. IV) GO TO 40
40              IOKVAL = .FALSE.
41              RETURN
42    35     IF (IVALUE .GT. IV) GO TO 40
43              IOKVAL = .FALSE.
44              RETURN
      C
      C  GET NEXT CONSTRAINT
45    40     IR = IRELN(JNEXT,IR)
46           GO TO 5
47           END
```

UTILITY ROUTINES - IRFIND FUNCTION

```
ISN
 1         FUNCTION IRFINC(IRECTP,ID,IVEH)
     C
     C        ROUTINE TO SEARCH FOR A COMPONENT RECORD HAVING A GIVEN ID
     C     STORED IN THE USER SPECIFICATION VEHICLE DESCRIPTION.  RETURNS A
     C     POINTER TO THIS RECORD OR NULL IF SUCH A RECORD DOESN'T EXIST.
     C
     C     INPUT PARAMETERS:
     C        IRECTP   RECORD TYPE
     C        ID       ID TO LOCK FOR
     C        IVEH     VEHICLE DESCRIPTION IN WHICH TO SEARCH FOR RECORD
     C
 2         COMMON /VEH/
 3         DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +      IRELN(3,50),VFILE(1),IVFILE(1)
 4         EQUIVALENCE  (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +      (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5         EQUIVALENCE  (VEHCLE(1),VFILE(1),IVFILE(1))
```

UTILITY ROUTINES - IRFIND FUNCTION

```
            COMMON /VPROTO/
            REAL*8 FLDNAM,RECNAM,DFAULT
            CCMMCN /JPARS/
            COMMON /JPARS1/
            CCMMCN /NPARS/
            COMMON /OUTPLT/
            LOGICAL*1 DF
            REAL*8 XNAT,XMANUF,XMODEL
            COMMON /AUX/
            REAL*8 BL8,CASFES
            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
            ICMP = IRECTP - 1
            IRFIND = IVECLE(ICMP,IVEH)
      C  ARE THERE ANY CCMPONENT RECORDS IN LIST ?
            IF (IRFIND .EQ. NULL) RETURN
            IOFF = IFILPT(IRECTP)
            NF = NFLDS(IRECTP)
      C  TEST WHETHER ID MATCHES THAT OF CURRENT RECORD
      10    KID = IVAL(VFILE(IOFF),NF,JID,IRFIND)
            IF (ID .EQ. KID) RETURN
      C  TRY NEXT RECCRD
            IRFIND = IVAL(VFILE(IOFF),NF,JNEXT,IRFIND)
            IF (IRFIND .NE. NULL) GO TO 10
            RETURN
            END
```

UTILITY ROUTINES - ITABLE FUNCTION

```
      FUNCTION ITABLE(LIST,KEY,NLIST,NKEY)
C
C        ROUTINE TO SEARCH A TABLE FOR A SPECIFIED KEY.  IF THE KEY IS
C     PRESENT THE ROUTINE RETURNS ITS INDEX IN THE TABLE.  IF THE ROUTINE
C     ENCOUNTERS A BLANK SLOT OF THE TABLE IT STORES THE KEY THERE AND
C     RETURNS ITS INDEX.  IF IT REACHES THE END OF THE TABLE BEFORE EITHER
C     FINDING THE KEY OR A BLANK SLOT, IT RETURNS A NULL VALUE.
C
C     INPUT PARAMETERS:
C        LIST    LIST OF ITEMS TO SEARCH
C        KEY     KEY TO LOOK FOR
C        NLIST   SIZE OF LIST
C        NKEY    NUMBER OF WORDS IN EACH ITEM IN LIST & IN KEY
C
      DIMENSION KEY(NKEY),LIST(NKEY,NLIST)
      DATA NULL/-1/,IBLS/' '/
C
      DO 20 I=1,NLIST
         DO 5 J=1,NKEY
            IF (KEY(J) .NE. LIST(J,I)) GO TO 10
    5       CONTINUE
         ITABLE = I
         RETURN
C
   10    DO 15 J=1,NKEY
            IF (LIST(J,I) .NE. IBLS) GO TO 20
   15       CONTINUE
         DO 18 J=1,NKEY
   18       LIST(J,I) = KEY(J)
         ITABLE = I
         RETURN
C
   20    CONTINUE
      ITABLE = NULL
      RETURN
```

```
ISN
20          END
```

UTILITY ROUTINES - IVAL FUNCTION

```
ISN
 1           FUNCTION IVAL(IFILE,NFLD,IFLD,IREC)
      C
      C        FUNCTION TO RETRIEVE AND RETURN THE VALUE OF A SPECIFIED
      C     INTEGER FIELD OF A SPECIFIED RECORD IN A SPECIFIED FILE (ARRAY).
      C
      C     INPUT PARAMETERS:
      C        IFILE    FILE IN WHICH VALUE IS LOCATED
      C        NFLD     NUMBER OF FIELDS IN EACH RECORD OF FILE
      C        IFLD     FIELD OF INTEREST
      C        IREC     RECORD OF INTEREST
      C
 2             DIMENSION IFILE(NFLD,1)
      C
 3             IVAL = IFILE(IFLD,IREC)
 4             RETURN
 5             END
```

UTILITY ROUTINES - IZERO SUBROUTINE

```
ISN
 1          SUBROUTINE IZERO(IARRAY,N)
     C
     C  INITIALIZES AN INTEGER ARRAY TO ZERO
     C
 2          DIMENSION IARRAY(N)
 3          DO 10 I=1,N
 4   10         IARRAY(I) = 0
 5          RETURN
 6          END
```

```
                         UTILITY ROUTINES - LSERCH FUNCTION
ISN
 1              FUNCTION LSERCH(LIST,KEY,NLIST,NKEY)
       C
       C           FUNCTION TO SEARCH A LIST (VECTOR) FOR A SPECIFIED KEY.  THE
       C        FUNCTION RETURNS TFE INDEX OF THE KEY IN THE LIST IF SUCCESSFUL, OR
       C        A NULL VALUE IF NOT.
       C
       C        INPUT PARAMETERS:
       C           LIST     LIST OF ITEMS TO SEARCH
       C           KEY      KEY TO LOOK FOR
       C           NLIST    SIZE OF LIST
       C           NKEY     NUMBER OF WORDS CONTAINED BY EACH ITEM OF LIST & BY KEY
       C
 2              DIMENSION KEY(NKEY),LIST(NKEY,NLIST)
 3              DATA NULL/-1/
       C
 4              DO 10 I=1,NLIST
 5                 DO 5 J=1,NKEY
 6                    IF (KEY(J) .NE. LIST(J,I)) GO TO 10
 7     5              CONTINUE
 8                 LSERCH = I
 9                 RETURN
10     10          CONTINUE
11              LSERCH = NULL
12              RETURN
13              END
```

UTILITY ROUTINES - NEWREC FUNCTION

```
ISN
 1          FUNCTION NEWREC(IRECTP)
      C
      C       ROUTINE FOR OBTAINING THE NEXT AVAILABLE RECORD OF THE
      C   SPECIFIED TYPE FROM THE AVAILABLE SPACE LIST AND RETURNING
      C   A POINTER TO IT.
      C
      C   INPUT PARAMETERS:
      C       IRECTP  RECORD TYPE OF INTEREST
      C
 2          COMMON /VPROTO/
 3          REAL*8 FLDNAM,RECNAM,DFAULT
 4          COMMON /NPARS/
      C
 5          IF (IRECTP .LE. 1 .OR. IRECTP .GT. NRECTP) CALL ERR(7,
           +     'NEWREC',IRECTP,0,0)
 6          IF (LSTREC(IRECTP) .GE. NRECS(IRECTP)) CALL ERR(8,'NEWREC',
           +     IRECTP,NRECS(IRECTP),0)
 7          LSTREC(IRECTP) = LSTREC(IRECTP) + 1
 8          NEWREC = LSTREC(IRECTP)
 9          RETURN
10          END
```

```
ISN
 1         FUNCTION NEWREL(IRECTP,IFLD,KREL,IVALUE,ID)
      C
      C       FUNCTION TO GET A NEW RELATION RECORD AND STORE IT AS
      C    THE VALUE (OR ONE OF A LIST OF VALUES) OF A SPECIFIED FIELD
      C    IN A SPECIFIED COMPONENT RECORD OF THE USER SPECIFICATION VEHICLE.
      C
      C    INPUT PARAMETERS
      C       IRECTP  RECORD TYPE
      C       IFLD    FIELD INDEX OF THE COMPONENT TYPE
      C       KREL    AN INTEGER FROM 1 TO 6 INDICATING THE TYPE OF RELATION
      C       IVALUE  THE VALUE FIELD OF THE RELATION
      C       ID      COMPONENT IDENTIFIER IF RELATION REFERS TO A COMPONENT
      C
 2         COMMON /VEH/
 3         DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +     IRELN(3,50),VFILE(1),IVFILE(1)
 4         EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
```

UTILITY ROUTINES - NEWREL FUNCTION

```
     +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
      EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
      COMMON /VPROTO/
      REAL*8 FLDNAM,RECNAM,DFAULT
      COMMON /JPARS/
      COMMON /JPARSL/
      COMMON /NPARS/
      COMMON /AUX/
      REAL*8 BL8,DASHES
      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
C
C  GET A NEW RELATION RECORD, IF THERE IS ONE.
      IF (LSTREC(NRECTP) .GE. NRECS(NRECTP)) CALL ERR(30,'NEWREL',
     +     LSTREC(NRECTP),0,0)
      LSTREC(NRECTP) = LSTREC(NRECTP) + 1
      NEWREL = LSTREC(NRECTP)
      IOFF = IFILPT(IRECTP)
      NF = NFLDS(IRECTP)
      ICMP = IRECTP - 1
      IREC = ISPECV
      IF (IRECTP .EQ. 1) GO TO 10
         IREC = IVECLE(ICMP,ISPECV)
    2    IF (IREC .NE. NULL) GO TO 5
C  NO COMPONENT RECORD OF THIS TYPE MATCHING THE APPROPRIATE ID
C  CURRENTLY EXISTS FOR THE USER
C  SPECIFICATION VEHICLE.  CREATE ONE TO ATTACH RELATION RECORD TO.
            IREC = NEWREC(IRECTP)
            CALL STORE(IVECLE(ICMP,ISPECV),VFILE(IOFF),NF,JNEXT,IREC)
            IVECLE(ICMP,ISPECV) = IREC
C  STORE COMPONENT RECORD ID
            CALL STORE(ID,VFILE(IOFF),NF,JID,IREC)
            GO TO 15
C
C  IS THIS RECORD APPROPRIATE FOR ATTACHING CONSTRAINT TO ?
    5    KID = IVAL(VFILE(IOFF),NF,JID,IREC)
         IF (KID .NE. ID) GO TO 8
```

UTILITY ROUTINES - NEWREL FUNCTION

```
C  IF SO, CREATE A NEW RELATION RECORD AND STORE THE POINTER IN THE
C  APPROPRIATE FIELD.  IF A PREDEFINED ALTERNATE COMPONENT IS TO BE
C  CHANGED, WARN THE USER.
               IF (ID .GT. NALTCS(ICMP)) GO TO 15
               CALL WARN(1,'NEWREL',IRECTP,IFLD,ID)
               CALL STORE(IVALUE,VFILE(IOFF),NF,IFLD,IREC)
               RETURN
C  IF NOT, TRY OTHER COMPONENT RECORDS
 8           IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
             GO TO 2
10        IRELN(JNEXT,NEWREL) = IVECLE(IFLD,ISPECV)
15        IF (IRECTP .NE. 1) IRELN(JNEXT,NEWREL) =
     +         IVFILE(IOFF + (NF * (IREC-1)) + (IFLD-1))
          CALL STORE(NEWREL,VFILE(IOFF),NF,IFLD,IREC)
          IRELN(JRELCP,NEWREL) = KREL
          IRELN(JVALUE,NEWREL) = IVALUE
          RETURN
          END
```

UTILITY ROUTINES - NEWVEH FUNCTION

```
ISN
 1            FUNCTION NEWVEH(I1,I2,I3,I4)
        C
        C     FUNCTION TO GET A NEW RECCRD IN IRECS AND STORE
        C  INFORMATICN IN IT.  THE FUNCTION RETURNS A POINTER TO THE
        C  RECORD OBTAINED.
        C
        C  INPUT PARAMETERS:
        C     I1      VALUE OF 1ST FIELD
        C     I2      VALUE OF 2ND FIELD
        C     I3      VALUE OF 3RD FIELD
        C     I4      VALUE OF 4TH FIELD
        C
 2            CCMMON /SPECS2/
 3            REAL*8 VEHS
        C
 4            LSTVEH = LSTVEH + 1
 5            IF (LSTVEH .GT. NVR) CALL ERR(21,'NEWVEH',NVR,I4,0)
 6            NEWVEH = LSTVEH
 7            IRECS(JNXT,LSTVEH) = I1
 8            IRECS(JTYP,LSTVEH) = I2
 9            IRECS(JSPEC,LSTVEH) = I3
10            IRECS(JVEH,LSTVEH) = I4
11            RETURN
12            END
```

```
                          UTILITY ROUTINES - NLIST FUNCTION
ISN
 1            FUNCTICN NLIST(IFILE,NF,JNEXT,ISAVE)
      C
      C       ROUTINE TO CCUNT THE NUMBER OF RECORDS CHAINED TOGETHER IN A LIST.
      C
      C   INPUT PARAMETERS:
      C       IFILE    FILE CONTAINING THE RECORDS
      C       NF       NUMBER CF FIELDS OF FILE
      C       JNEXT    NEXT POINTER
      C       ISAVE     1ST RECCRD OF LIST
      C
 2            CCMMON /AUX/
 3            REAL*8 BL8,DASHES
 4            EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
 5            DIMENSION IFILE(NF,1)
 6            IR=ISAVE
      C
 7            NLIST = 0
 8    5       IF (IR .EQ. NULL) RETURN
 9            IR = IFILE(JNEXT,IR)
10            NLIST = NLIST + 1
11            GO TO 5
12            END
```

```
                     UTILITY ROUTINES - NULIFY SUBROUTINE
ISN
 1            SUBROUTINE NULIFY(IBLOCK,N)
      C
      C       ROUTINE TO NULL OUT THE CONTENTS OF A BLOCK OF STORAGE.
      C
      C   INPUT PARAMETERES:
      C     IBLOCK    LOCATION OF THE BEGINNING OF THE BLOCK TO BE NULLED.
      C     N         SIZE OF THE BLOCK TO BE NULLED
      C
 2            DIMENSION IBLOCK(N)
 3            DATA NULL/-1/
      C
 4            DO 10 I=1,N
 5     10        IBLOCK(I) = NULL
 6            RETURN
 7            END
```

UTILITY ROUTINES - FUNCTION OKVAL

```
ISN
 1          LOGICAL FUNCTION OKVAL(VAL,IRECTP,IFLD,ISAV)
      C
      C        FUNCTION TO TEST WHETHER THE USER HAS SPECIFIED ANY CONSTRAINTS
      C     ON A PARTICULAR ATTRIBUTE'S VALUE AND IF SO WHETHER THE SPECIFIED
      C     VALUE SATISFIES THESE CONSTRAINTS.  THE FUNCTION RETURNS .TRUE. IF NO
      C     CONSTRAINTS ARE SPECIFIED OR IF THE INDICATED VALUE SATISFIES ALL
      C     CONSTRAINTS.
      C
      C     INPUT PARAMETERS:
      C        VAL      VALUE TO BE TESTED
      C        IRECTP   RECORD TYPE IN WHICH VALUE IS LOCATED
      C        IFLD     FIELD OF RECORD IN WHICH VALUE LOCATED
      C       ISAV     RECORC NUMBER IF RELATION RECORD PASSED
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +     IELECT(26,6),ICOMMO(28,9),IFIRFX(28,10),IENVIR(26,9),
          +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
```

UTILITY ROUTINES - FUNCTION OKVAL

```
ISN
           +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
           +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
  5             EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
  6             COMMON /VPROTO/
  7             REAL*8 FLDNAM,RECNAM,DFAULT
  8             COMMON /JPARS/
  9             COMMON /JPARS1/
 10             COMMON /AUX/
 11             REAL*8 BL8,DASHES
 12             EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
 13             OKVAL = .TRUE.
      C
      C  GET CONSTRAINT RECORD IF ONE EXISTS
 14             NF = NFLDS(IRECTP)
 15             IOFF = IFILPT(IRECTP)
      C
      C   TEST IF RELATION REFERENCE PASSED
      C
 16             IF(ISAV .EQ. NULL) GO TO 3
 17             IREC=ISAV
 18             GO TO 4
 19   3         IREC = ISPECV
 20             IF (IRECTP .GT. 1) IREC = IVECLE(IRECTP-1,ISPECV)
 21             IF (IREC .EQ. NULL) RETURN
 22   4         IR = IVAL(VFILE(IOFF),NF,IFLD,IREC)
 23   5         IF (IR .EQ. NULL) RETURN
 24             V = RELN(JVALUE,IR)
 25             IRTYP = IRELN(JRELOP,IR)
 26             GO TO (10,15,20,25,30,35),IRTYP
 27   10        IF (VAL .LT. V) GO TO 40
 28                OKVAL = .FALSE.
 29                RETURN
 30   15        IF (VAL .LE. V) GO TO 40
 31                OKVAL = .FALSE.
 32                RETURN
```

UTILITY ROUTINES - FUNCTION OKVAL

```
ISN
33    20     IF (VAL .EQ. V) GO TO 40
34              OKVAL = .FALSE.
35              RETURN
36    25     IF (VAL .NE. V) GO TO 40
37              OKVAL = .FALSE.
38              RETURN
39    30     IF (VAL .GE. V) GO TO 40
40              OKVAL = .FALSE.
41              RETURN
42    35     IF (VAL .GT. V) GO TO 40
43              OKVAL = .FALSE.
44              RETURN
      C
      C  GET NEXT CONSTRAINT
45    40     IR = IRELN(JNEXT,IR)
46           GO TO 5
47           END
```

UTILITY ROUTINES - QUEUE SUBROUTINE

```
ISN
 1           SUBROUTINE QLEUE(IR,LISTPT,IRECTP)
      C
      C      ROUTINE FOR QUEUEING A RECCRD ONTO A LIST.
      C
      C   INPUT PARAMETERS:
      C      IR        THE RECORD TO BE QUEUED
      C      LISTPT    A POINTER TO THE HEAD OF THE LIST
      C      IRECTP    THE TYPE OF RECORD
      C
 2           COMMON /VEH/
 3           DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
           +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +      IELECT(26,6),ICCMMC(28,9),IFIREX(28,10),IENVIR(26,9),
           +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +      IRELN(3,50),VFILE(1),IVFILE(1)
 4           EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +      ISPFNG(1)),(TPACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
           +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
           +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
           +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
           +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
           +      (SMCKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5           EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6           COMMON /VPROTO/
 7           REAL*8 FLDNAM,RECNAM,DFAULT
```

UTILITY ROUTINES - QUEUE SUBROUTINE

```
          CCMMON /JPARS/
          COMMON /JPARS1/
          CCMMON /AUX/
          REAL*8 BL8,DASHES
          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
C
          IOFF = IFILPT(IRECTP)
          NF = NFLDS(IRECTP)
          IF (LISTPT .NE. NULL) GO TO 10
             LISTPT = IR
             CALL STORE(NULL,VFILE(IOFF),NF,JNEXT,IR)
             RETURN
C
       10 IREC = LISTPT
C      GET VALUE OF NEXT FIELD OF RECORD
   15     IRECL = IREC
          IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
          IF (IREC .NE. NULL) GO TO 15
          CALL STORE(IR,VFILE(IOFF),NF,JNEXT,IRECL)
          CALL STORE(NULL,VFILE(IOFF),NF,JNEXT,IR)
          RETURN
          END
```

UTILITY ROUTINES - STCNAM SUBROUTINE

```
ISN
 1            SUBROUTINE STONAM(ONAME,IVFLAG)
      C
      C          ROUTINE TO STORE AN OUTPUT NAME READ FROM THE PROTOTYPE
      C   DEFINITION FILE IN THE NEXT AVAILABLE RECORD OF THE OJTNAM ARRAY.
      C
      C   INPUT PARAMETERS:
      C      ONAME    NAME TO BE STORED
      C      IVFLAG   AN INDICATOR WHICH HAS THE VALUE (O) IF THERE IS NO VALUE
      C               TO PRINT. IF THERE IS A VALUE TO PRINT THEN IFLDRC IS THE
      C               INDEX OF THE FIELD IN THE RECORD.
      C
 2            COMMON /VPROTO/
 3            REAL*8 FLDNAM,RECNAM,DFAULT
 4            COMMON /NPARS/
      C
 5            DIMENSION ONAME(NOUTWD)
      C
 6            LSTOUT = LSTOUT + 1
 7            IF (LSTOUT .GT. NOUT) CALL ERR(31,'STONAM',NOUT,0,0)
 8            CALL COPY(ONAME,OUTNAM(1,LSTOUT),NOUTWD)
 9            IOUTF(LSTOUT) = IVFLAG
10            RETURN
11            END
```

UTILITY ROUTINES - STORE SUBROUTINE

```
ISN
 1          SUBROUTINE STORE(VALUE,FILE,NFLD,IFLD,IREC)
      C
      C        ROUTINE TO STORE A VALUE IN A PARTICULAR FIELD OF A
      C     PARTICULAR RECORD OF A SPECIFIED FILE (ARRAY).
      C
      C     INPUT PARAMETERS:
      C        VALUE   VALUE TO BE STORED
      C        FILE    FILE IN WHICH TO STORE VALUE
      C        NFLD    NUMBER OF FIELDS IN EACH RECORD OF FILE
      C        IFLD    FIELD IN WHICH TO STORE VALUE
      C        IREC    RECORD IN WHICH TO STORE VALUE
      C
 2          DIMENSION FILE(NFLD,1)
      C
 3          FILE(IFLD,IREC) = VALUE
 4          RETURN
 5          END
```

```
                          UTILITY ROUTINES - SUM FUNCTION
ISN
 1              FUNCTICN SUM(X,N)
      C
      C   FUNCTION TO SUM THE ELEMENTS CF AN ARRAY.  FUNCTION RETURNS
      C  THIS SUM.
      C
 2              DIMENSION X(N)
 3              SUM = 0.
 4              DO 10 I=1,N
 5    10            SUM = SUM + X(I)
 6              RETURN
 7              END
```

```
                 UTILITY ROUTINES - SUMCMP FUNCTION
ISN
 1            FUNCTION SUMCMP(IV,IFLD,UNIT)
       C
       C      FUNCTION TO SUM A COMMON ATTRIBUTE SUCH AS WEIGHT, VOLUME, OR
       C   COST ACROSS ALL COMPONENTS OF A SPECIFIED VEHICLE.
       C
       C   INPUT PARAMETERS:
       C      IV       VEHICLE
       C      IFLD     FIELD OF RECORD TO SUM
       C      UNIT     UNITS OF MEASURE DESIRED
       C
 2            COMMON /VEH/
 3            DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
           +       FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR( 28,7),
           +       ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
           +       IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
           +       IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
           +       IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
           +       IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
           +       IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
           +       IRELN(3,50),VFILE(1),IVFILE(1)
 4            EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
           +       ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
           +       (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
           +       IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
           +       (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
           +       IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
           +       (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
           +       ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
           +       (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
           +       ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
           +       (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
           +       IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
           +       (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5            EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6            COMMON /VPROTO/
```

UTILITY ROUTINES - SUMCMP FUNCTION

```
      REAL*8 FLDNAM,RECNAM,DFAULT
      COMMON /JPARS/
      COMMON /JPARS1/
      COMMON /NPARS/
      COMMON /AUX/
      REAL*8 BLB,DASHES
      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
C
      SUMCMP = 0.
      DO 20 ICMP=1,NCMPTP
         IRECTP =  ICMP + 1
         NF = NFLDS(IRECTP)
         IOFF =  IFILPT(IRECTP)
         IREC = IVECLE(ICMP,IV)
   10    IF (IREC .EQ. NULL) GO TO 20
         SUMCMP = SUMCMP + DVAL(VFILE(IOFF),NF,IFLD,IREC,IRECTP,UNIT,DF)
         IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
         GO TO 10
   20    CONTINUE
      RETURN
      END
```

UTILITY ROUTINES - SUMPD2 FUNCTION

```
ISN
 1            FUNCTICN SUMFD2(IX,Y,N)
       C
       C   FUNCTION TO CALCULATE THE SUM OF THE PRODUCT OF CORRESPONDING
       C  ELEMENTS IN TWO EQUAL SIZED ARRAYS.
       C
       C
       C    COMPUTES TFE VECTOR PRODUCT IX*Y.
       C
 2            DIMENSION IX(N),Y(N)
 3            SIIMPD2 = 0.
 4            DO 10 I=1,N
 5     10         SUMPD2 = SUMPD2 + IX(I)*Y(I)
 6            RETURN
 7            END
```

```
                          UTILITY ROUTINES - SUMPRD FUNCTION
ISN
 1                FUNCTION SUMPRD(X,Y,N)
      C
      C   FUNCTION TO CALCULATE THE SUM OF THE PRODUCT OF CORRESPONDING
      C  ELEMENTS IN TWO EQUAL SIZED ARRAYS.
      C
      C
      C    COMPUTES THE VECTOR PRODUCT X*Y.
      C
 2                DIMENSION X(N),Y(N)
 3                SUMPRD = 0.
 4                DO 10 I=1,N
 5    10              SUMPRD = SUMPRD + X(I)*Y(I)
 6                RETURN
 7                END
```

UTILITY ROUTINES - TRANSF FUNCTION

```
ISN
 1          FUNCTION TRANSF(VALUE,MULOP,FACTOR,ADDOP,TERM)
      C
      C        ROUTINE TO TRANSFOPM A VALUE BY MULTIPLYING IT BE A CONSTANT
      C     AND/OR BY ADDING OR SUBTRACTING A CONTANT FROM IT, IF SUCH
      C     OPERATIONS ARE SPECIFIED BY THE USER.  THE FUNCTION RETURNS THE
      C     TRANSFORMED VALUE.
      C
      C     INPUT PARAMETERS:
      C        VALUE    VALUE TO BE TRANSFORMED (POSSIBLY)
      C        MULOP    MULTIPLY OPERATOR (OR BLANK)
      C        FACTOR   CONSTANT TO BE MULTIPLIED
      C        ADDOP    + OR - OPERATOR (OR BLANK)
      C        TERM     CONSTANT TO BE ADDED OR SUBTRACTED
      C
 2          COMMON /AUX/
 3          REAL*8 BL8,DASHES
 4          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
 5          TRANSF = VALUE
 6          IF (MULOP .NE. MULT) GO TO 10
 7             TRANSF = TRANSF * FACTOR
      C
 8    10    IF (ADDOP .NE. PLUS) GO TO 20
 9             TRANSF = TRANSF + TERM
10             RETURN
      C
11    20    IF (ADDOP .NE. SMINUS) RETURN
12             TRANSF = TRANSF - TERM
13             RETURN
14          END
```

```
ISN
 1          SUBROUTINE UNHOCK(ICMP)
      C
      C        ROUTINE TO UNCHAIN COMPONENT RECORDS OF A GIVEN TYPE CHAINED
      C     TOGETHER AS PART OF THE SPECIFICATION VEHCILE DESCRIPTION WHENEVER
      C     BACKTRACKING IS NECESSARY.
      C
      C     INPUT PARAMETERS:
      C        ICMP     COMPONENT TYPE
      C
 2          COMMON /VEH/
 3          DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
          +     FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
          +     ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
          +     IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
          +     IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
          +     IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
          +     IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
          +     IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
          +     IRELN(3,50),VFILE(1),IVFILE(1)
 4          EQUIVALENCE (VEHCLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
          +     ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
          +     (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
          +     IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
          +     (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
          +     IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
          +     (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
          +     ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
          +     (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
          +     ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
          +     (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
          +     IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
          +     (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
 5          EQUIVALENCE (VEHCLE(1),VFILE(1),IVFILE(1))
 6          COMMON /VPROTO/
 7          REAL*8 FLDNAM,RECNAM,DFAULT
```

UTILITY ROUTINES - UNHOOK SUBROUTINE

```
ISN
 8          CCMMON /JPARS/
 9          COMMON /JPARSL/
10          COMMON /AUX/
11          REAL*8 BL8,CASHES
12          EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      C
13          IR = IVECLE(ICMP,ISPECV)
14          IF (IR .EQ. NULL) RETURN
15          IVECLE(ICMP,ISPECV) = NULL
16          IRECTP = ICMP + 1
17          NF = NFLDS(IRECTP)
18          IOFF = IFILPT(IRECTP)
19    10    NEXT = IVAL(VFILE(IOFF),NF,JNEXT,IR)
20          CALL STORE(NLLL,VFILE(IOFF),NF,JNEXT,IR)
21          IF (NEXT .EQ. NULL) RETURN
22          IR = NEXT
23          GO TO 10
24          END
```

UTILITY ROUTINES - ZERO SUBROUTINE

```
I SN
 1            SUBROUTINE ZERO(V,N)
      C
      C       INITIALIZES AN ARRAY TO ZERO
      C
 2            DIMENSION V(N)
 3            DO 10 I=1,N
 4    10         V(I) = 0.
 5            RETURN
 6            END
```

DISTRIBUTION LIST

DISTRIBUTION LIST

Please notify USATARADCOM, DROTA-ZC, Warren, MI 48090, of corrections and/or changes in address.

No. of
Copies

13
Commander
US Army Tank-Automotive Research and Development Command
Attention: DRDTA-UL
Warren, MI 48090

17
Commander
US Army Tank-Automotive Research and Development Command
Attention: DRDTA-Z
Warren, MI 48090

3
Director
US Army Material Systems Analysis Activity
Attention: DRXSY-CM (Messrs. D. Woomert, W. Niemeyer, W. Criswell)
Aberdeen Proving Grounds, MD 21005

1
Director
US Army Human Engineering Laboratory
Attention: Mr. A. Eckels
Aberdeen Proving Grounds, MD 21005

2
Director
US Army Ballistic Research Laboratory
Attention: Dr. A. Rainis, Mr. G. Bowers
Aberdeen Proving Grounds, MD 21005

2
Director
Defense Documentation Center
Cameron Station
Alexandria, VA 22314

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 12444	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PARAMETRIC ENGINEERING SYSTEM DEFINITION MODEL VOLUME I--MAIN REPORT, APPENDICES A & B VOLUME II--APPENDIX C		5. TYPE OF REPORT & PERIOD COVERED Final Report, 4/19/78 - 9/29/78 6. PERFORMING ORG. REPORT NUMBER VRI-TARADCOM-1 FR78-1
7. AUTHOR(s) S. Spaulding, A. Weintraub, F. Cioch, J. Lenz		8. CONTRACT OR GRANT NUMBER(s) DAAK30-78-C-0059
9. PERFORMING ORGANIZATION NAME AND ADDRESS Vector Research, Incorporated P.O. Box 1506 Ann Arbor, Michigan 48106		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Tank-Automotive Research and Development Command, Attn: DRDTA-PAV Warren, Michigan 48090		12. REPORT DATE August 1979 13. NUMBER OF PAGES 360
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) US Army Tank-Automotive Research and Development Command, Attn: DRDTA-NK Warren, Michigan 48090		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release;
distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

N/A

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Armored Combat Vehicles
Tanks
System Performance Estimation
Parametric Design Models
R&D Planning Models

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a parametric engineering system definition model developed for use in planning tank-automotive research and development. The model can be used for (1) estimating the performance of a conceptual armored combat vehicle consisting of a specified set of components or (2) "sizing" a vehicle to meet a set of performance specifications. Key elements of the model include a structure for a data base to contain descriptions of components which might be incorporated into future armored

combat vehicles, (2) a solution algorithm which uses a combinatorial approach to search over alternative combinations of components to find one which meets specifications input by the model user, (3) a variety of engineering relationships and "look-up table" functions for estimating system engineering parameters and performance characteristics, and (4) routines which output a description of the concept vehicle generated by the model.